ระบบผู้ให้บริการประทับเวลา

TIME-STAMP AUTHORITY SERVICE

โดย

ปฏิเวธ เรืองสุวรรณ

PATIVETH RUANGSUWAN

อาจารย์ที่ปรึกษา

ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2549

TIME-STAMP AUTHORITY SERVICE

PATIVETH RUANGSUWAN

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2006

ชื่อหัวข้อ	การพัฒนาระบบให้บริการประทับเวลา
นักศึกษา	นายปฏิเวธ เรืองสุวรรณ
รหัสนักศึกษา	48066529
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

บทคัดย่อ

ในหลายสถานการณ์ความถูกต้องของเวลาที่ระบุในข้อมูลอิเล็กทรอนิกส์มีความจำเป็นสำหรับการทำธุรกรรมอิเล็กทรอนิกส์เหล่านั้น ซึ่งการระบุว่าข้อมูลอิเล็กทรอนิกส์นั้นถูกสร้างขึ้นหรือมีการแก้ไขเมื่อใดจะมีการระบุเวลาโดยอ้างอิงจากแหล่งเวลาเฉพาะที่ (local time) แต่เนื่องจากเวลาเฉพาะที่นั้นไม่มีการรับรองความน่าเชื่อถือ และสามารถที่จะถูกแก้ไขได้โดยง่าย จึงไม่สามารถนำเวลาจากแหล่งเวลาเฉพาะที่มาใช้ยืนยัน หรืออ้างอิงได้ ดังนั้นแนวทางในการแก้ปัญหาดังกล่าวสามารถใช้การประทับเวลาซึ่งเป็นเทคนิคที่ทำให้สามารถพิสูจน์และยืนยันได้ว่า ข้อมูลอิเล็กทรอนิกส์นั้น ได้ถูกสร้างขึ้นหรือมีการแก้ไขเมื่อใด โดยอ้างอิงจากแหล่งเวลาที่มีความน่าเชื่อถือ ซึ่งในโครงการนี้ได้นำแนวคิดและกระบวนการประทับเวลาตามมาตรฐาน RFC 3161 (Internet X.509 Public Key Infrastructure: Time-Stamp Protocol) โดยมีการใช้ใบรับรองอิเล็กทรอนิกส์ที่สอดคล้องตามมาตรฐาน RFC 3280 (Internet X.509 Public Key Infrastructure: Certificate and CRL Profile) มาพัฒนาระบบผู้ให้บริการประทับเวลา ทั้งนี้ผลของการประทับเวลาโดยวิธีการดังกล่าวสามารถใช้ยืนยัน หรือตรวจสอบความถูกต้องได้ในภายหลัง ซึ่งจะเพิ่มความน่าเชื่อถือให้กับธุรกรรมอิเล็กทรอนิกส์เหล่านั้นต่อไป

Title	Time-Stamp Authority Service
Student	Mr. Pativeth Ruangsuwan
Student ID.	48066529
Degree	Master of Science
Programme	Information Science
Academic Year	2006
Advisor	Asst.Prof. Dr.Chanboon Sathitwiriyawong

ABSTRACT

In some situations there is a need to prove the precise time for the electronic document when it has been created or last modified, especially when performing electronic transactions. To specify the time in creation or modification of electronic documents we usually refer to the local time. However, local time is easy to modify and it is not reliable. As a result, we can't trust the time that is specified by local time. The solution for this problem is Time-Stamping. It is the process or technique that is designed to prove the exact time when electronic document was created or modified, using the time from trusted source. This project has adopted the concept from RFC 3161 (Internet X.509 Public Key Infrastructure: Time-Stamp Protocol) for Time-Stamping and RFC 3280 (Internet X.509 Public Key Infrastructure: Certificate and CRL Profile) for digital certificate service. The outcome for the Time-Stamping process can be used to confirm and prove the exact time for electronic documents in the future, and also improve the trust for electronic transactions.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงาน เรื่อง การพัฒนาระบบผู้ให้บริการประทับเวลา (Time-Stamp Authority Service) สำเร็จลงได้ด้วยความอนุเคราะห์จากบุคคลหลายฝ่าย ผู้เขียนใคร่ขอแสดงความระลึกถึงบุคคลสำคัญผู้ให้ความกรุณาดังต่อไปนี้

ขอกราบขอบพระคุณ ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ อาจารย์ที่ปรึกษาโครงงานและอาจารย์ประจำภาควิชาเทคโนโลยีสารสนเทศเป็นอย่างยิ่ง ที่กรุณาให้โอกาสในการทำโครงงานนี้ ตลอดจนการให้ความอนุเคราะห์ ให้คำแนะนำต่าง ๆ ทำให้การทำโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณสำนักบริการเทคโนโลยีสารสนเทศภาครัฐ ในการอนุเคราะห์ความรู้ทางด้าน PKI ให้แก่ข้าพเจ้า เพื่อเป็นพื้นฐานในการทำโครงงานฉบับนี้ให้สำเร็จ

ขอขอบคุณบัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะเทคโนโลยีสารสนเทศที่ให้ความช่วยเหลือในเรื่องต่างๆ

ขอขอบคุณเพื่อน ๆ หลักสูตรวิทยาศาสตรมาบัณฑิต สาขาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำและให้ความช่วยเหลือต่าง ๆ เป็นอย่างดี รวมทั้งคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา ครอบครัวและเพื่อน ๆ ของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำโครงงานฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากโครงงานฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ปฏิเวธ เรืองสุวรรณ

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมา

ลายมือชื่อดิจิทัล (Digital Signature) เป็นข้อมูลอิเล็กทรอนิกส์ชุดหนึ่งซึ่งถูกสร้างขึ้นโดยใช้เทคโนโลยีระบบรหัส อันเป็นพื้นฐานสำคัญของโครงสร้างพื้นฐานระบบกุญแจสาธารณะ (Public Key Infrastructure: PKI) มีวัตถุประสงค์เพื่อเชื่อมโยงความสัมพันธ์ระหว่างผู้สร้างลายมือชื่อเข้ากับข้อมูลอิเล็กทรอนิกส์ใดๆ อันจะทำให้สามารถเชื่อมั่นได้ว่าบุคคลนั้นเป็นผู้สร้างลายมือชื่อเพื่อรับรองข้อมูลนั้นจริง รวมทั้งสามารถพิสูจน์ความถูกต้องครบถ้วนของข้อมูลอิเล็กทรอนิกส์เหล่านั้นได้ อย่างไรก็ตาม จากคุณสมบัติของลายมือชื่อดิจิทัล ไม่ได้มีการยืนยันถึงเวลาที่ใช้ในการสร้างลายมือชื่อดิจิทัล ทำให้อาจเป็นปัญหาได้ว่าสิ่งใดจะสามารถพิสูจน์เวลาที่สร้างลายมือชื่อดิจิทัลได้ ทั้งนี้เนื่องจากไม่มีข้อกำหนดใดยืนยันความถูกต้องของเวลาที่ใช้ในการสร้างลายมือชื่อ ซึ่งปัญหาดังกล่าวสามารถแก้ไขได้โดยพิจารณาเวลาจากแหล่งอื่นที่มีความน่าเชื่อถือ หรือให้มีการประทับเวลา (Time-Stamping) ลงบนลายมือชื่อดิจิทัลดังกล่าว

มีหลายสถานการณ์ที่มีความต้องการทราบว่าเอกสารหนึ่งๆ ถูกสร้างขึ้น หรือแก้ไขครั้งล่าสุดเมื่อใด เช่น ในประเด็นเกี่ยวกับทรัพย์สินทางปัญญา ซึ่งต้องการทราบอย่างแน่ชัดว่า ผู้ค้นพบได้บันทึกทรัพย์สินทางปัญญานั้นในรูปแบบอิเล็กทรอนิกส์ลงในสื่อที่สามารถจดทะเบียนได้เมื่อใด เพื่อที่จะสามารถเปรียบเทียบลำดับการค้นพบก่อนหลังได้ วิธีการที่สามารถนำมาแก้ไขปัญหาดังกล่าวได้คือ การประทับเวลา อย่างไรก็ตาม การประทับเวลาไม่สามารถบ่งชี้ได้อย่างแน่ชัดว่าข้อมูลอิเล็กทรอนิกส์ที่ได้รับการประทับเวลาถูกสร้างขึ้นหรือแก้ไขครั้งล่าสุดเมื่อใด เนื่องจากการประทับเวลาจะเกิดขึ้นภายหลังจากเวลาที่ถูกสร้างหรือแก้ไขเสมอ การประทับเวลาสามารถยืนยันได้เพียงว่า ข้อมูลอิเล็กทรอนิกส์นั้น ได้ถูกสร้างขึ้นและมีอยู่ก่อนเวลาที่ได้มีการประทับเวลาไว้

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อพัฒนาระบบที่ใช้ในการตรวจสอบลายมือชื่อดิจิทัลได้อย่างถูกต้อง
1.2.2 เพื่อให้ลายมือชื่อดิจิทัลสามารถเป็นหลักฐานในการทำธุรกรรมอิเล็กทรอนิกส์ได้อย่างถูกต้อง
1.2.3 เพื่อให้ลายมือชื่อดิจิทัลมีความถูกต้องในระยะยาว
1.2.4 เพื่อให้การทำธุรกรรมอิเล็กทรอนิกส์มีประสิทธิภาพเทียบเท่ากับการทำธุรกรรมที่มีเอกสาร

1.3 ขอบเขตของโครงการ

การพัฒนาโครงการระบบระบบประทำเวลาแบบไม่ต่อเนื่อง แบ่งการพัฒนาระบบออกเป็น 3 ส่วนหลัก ๆ คือ

1.3.1 ส่วนในการสร้าง Time-Stamp สำหรับลายมือชื่อดิจิทัล

1.3.2 ส่วนในการตรวจสอบ Time-Stamp ที่ออกโดยระบบ

1.3.3 ส่วนการจัดการ Time-Stamp เพื่อใช้สำหรับค้นหา

ซึ่งในการพัฒนาอยู่บนสมมุติฐานว่าระบบผู้ให้บริการประทับเวลานี้อยู่ในพื้นที่ปลอดภัย และมีเฉพาะผู้ที่มีสิทธิ์เท่านั้นที่สามารถเข้าถึงได้ ซึ่งเน้นเฉพาะการพัฒนาระบบเพื่อให้บริการ แต่จะไม่ได้ลงลายละเอียดในส่วนการบริหารจัดการในส่วนการให้บริการ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ระบบที่เป็นต้นแบบในการพัฒนาระบบประทับเวลาต่อไป

1.4.2 ยกระดับมาตรฐานของการทำธุรกรรมอิเล็กทรอนิกส์ให้มีประสิทธิภาพมากขึ้นและสามารถ

1.5 ขั้นตอนในการพัฒนาระบบ

ตารางที่ 1.1 ขั้นตอนในการพัฒนาระบบ

ลำดับ	รายละเอียด	ปี 2549			ปี 2550		
		ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1	รวบรวมรายละเอียดและศึกษาความเป็นไปได้ของโครงการ						
2	กำหนดขอบเขตและรวบรวมความต้องการของระบบ (Requirement)						
3	ศึกษาทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบ						
4	วิเคราะห์และออกแบบระบบ (Analysis & Design)						
	4.1 ออกแบบแอพพลิเคชั่น (Application Design)						
	4.2 ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้ (User Interface Design)						
5	พัฒนาระบบ (Implementation)						
	5.1 พัฒนา Module Generate Time-Stamp						
	5.2 พัฒนา Module การจัดการ Time-Stamp						
	5.3 พัฒนา Module Application ในการทำ Time-Stamp Request						
6	ทดสอบระบบและทำการแก้ไข (Test and Debug)						
7	จัดทำเอกสารโครงการ						

บทที่ 2
ทฤษฎีและหลักการที่ใช้ในการพัฒนาระบบงาน

2.1 เทคโนโลยีระบบรหัส (Cryptography Technology)

การทำให้ข้อมูลอยู่ในรูปแบบที่ไม่สามารถอ่านออกได้ เรียกว่า การเข้ารหัส (Encryption) เพื่อทำให้ข้อมูลนั้นเป็นความลับ ซึ่งผู้ที่มีสิทธิ์เท่านั้นที่จะสามารถอ่านข้อมูลนั้นได้ ด้วยการถอดรหัส (Decryption) นั่นคือ สามารถรักษาข้อมูลให้เป็นความลับ (Confidentiality) และ กำหนดผู้มีสิทธิ์ได้ (Authentication & Authorization) สำหรับการเข้ารหัส และ ถอดรหัสนั้นจะอาศัยสมการทาง คณิตศาสตร์ที่ซับซ้อน และ ต้องอาศัยกุญแจซึ่งอยู่ในรูปของพารามิเตอร์ที่กำหนดไว้ (สำหรับตัวกุญแจนั้นจะมีความยาวเป็น บิต(bit) ยิ่งกุญแจมีความยาวมาก ก็จะทำให้มีความปลอดภัยมากขึ้น เนื่องจากจะต้องใช้เวลานานมากขึ้นในการ คาดเดากุญแจโดยผู้คุกคาม) ในการเข้า และ ถอดรหัส สามารถแบ่งออกเป็น 2 ประเภท คือ ระบบกุญแจแบบสมมาตร(Symmetric Key Cryptography หรือ Secret Key Cryptography) และระบบกุญแจแบบอสมมาตร (Asymmetric Key Cryptography หรือ Public Key Cryptography)

จุดประสงค์ของระบบรหัส

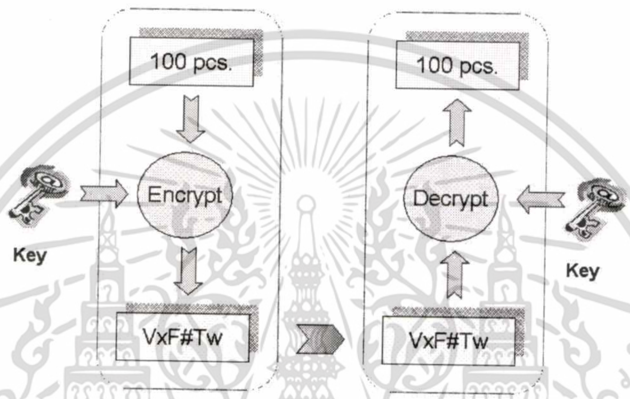
1. การรักษาความลับ(Confidentiality) เพื่อป้องกันไม่ให้ผู้ที่ไม่มีสิทธิ์ในการเข้าถึงข้อมูลสามารถเข้าถึงข้อมูลได้
2. ความครบถ้วนของข้อมูล(Data Integrity) เพื่อป้องกันข้อมูลให้อยู่ในสภาพเดิมอย่างสมบูรณ์ กล่าวคือ ในกระบวนการสื่อสารนั้นผู้รับ (Receiver) ได้รับข้อมูลที่ถูกต้องตามที่ผู้ส่ง (Sender) ส่งมาให้โดยข้อมูลจะต้องไม่มีการสูญหายหรือถูกเปลี่ยนแปลงแก้ไขใดๆ
3. การพิสูจน์ตัวจริง(Authentication) เพื่อระบุตัวตนที่แท้จริงของผู้ส่งข้อมูลอิเล็กทรอนิกส์
4. การห้ามปฏิเสธความลับผิด(Non-repudiation) การป้องกันไม่ให้บุคคลผู้ส่งปฏิเสธว่าตนไม่ได้ส่งข้อมูลอิเล็กทรอนิกส์

2.1.1 การเข้ารหัสแบบสมมาตร (Symmetric-Key Cryptography)

ระบบรหัสแบบสมมาตรเป็นระบบที่ใช้กุญแจชุดเดียวกันทั้งด้านผู้ส่งและผู้รับในการเข้าและถอดรหัสลับ ดังรูปที่ 1 จากสาเหตุนี้จึงเป็นที่มาของชื่อระบบรหัสแบบสมมาตร เนื่องจากคำว่า

“สมมาตร” เป็นการอธิบายถึงความเท่ากันหรือเหมือนกันของสองข้าง ซึ่งในที่นี้ก็คือตัวกุญแจนั่นเอง

กุญแจซึ่งอยู่ในรูปรหัสคอมพิวเตอร์นี้ เป็นตัวแปรสำคัญสำหรับการเข้าและถอดรหัสลับข้อมูล ซึ่งขนาดของกุญแจ (มีหน่วยเป็นบิต : bit) จะแสดงถึงระดับความปลอดภัยของข้อมูลที่ได้รับการเข้ารหัสลับ โดยการใช้กุญแจที่มีความยาวหรือจำนวนบิตสูง จะทำให้การเข้ารหัสลับข้อมูลนั้นมีความปลอดภัยมากยิ่งขึ้น สำหรับขนาดของกุญแจที่นิยมใช้ในระบบรหัสแบบสมมาตร คือ 40 – 512 บิต ตัวอย่างวิธีการเข้า/ถอดรหัสลับของระบบรหัสแบบสมมาตร เช่น DES, 3DES, AES, CAST, RC2 เป็นต้น



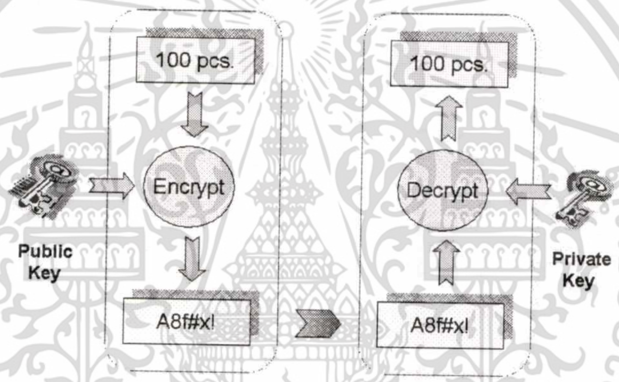
รูปที่ 2.1 การเข้า/ถอดรหัสแบบสมมาตร

ข้อดีของระบบรหัสแบบสมมาตร	ข้อเสียของระบบแบบสมมาตร
- สามารถเข้า/ถอดรหัสลับได้อย่างรวดเร็วกว่าระบบกุญแจแบบอสมมาตร 100 -1000 เท่า - ใช้ในการเก็บรักษาความลับของข้อมูล	- การบริหารจัดการกุญแจ (Key Management) ทำได้ยาก เนื่องจากผู้ใช้ต้องใช้กุญแจ 1 ดอก สำหรับติดต่อกับบุคคล 1 คน ดังนั้นหากติดต่อกับบุคคลจำนวนมาก ก็จะต้องเก็บกุญแจเพิ่มขึ้นตามจำนวนผู้ที่ติดต่อ - การส่งกุญแจไปยังบุคคลที่ติดต่อด้วยนั้น มีความเสี่ยงในการที่ผู้อื่นอาจลักลอบขโมยกุญแจไประหว่างการ ส่งกุญแจดังกล่าว

2.1.2 การเข้ารหัสแบบอสมมาตร (Asymmetric Cryptography)

ระบบรหัสแบบอสมมาตรเป็นระบบที่ใช้กุญแจคู่ (Key Pair) ซึ่งประกอบด้วยกุญแจส่วนตัว (Private Key) และกุญแจสาธารณะ (Public Key) กล่าวคือ ผู้ใช้ 1 คนหรือที่เรียกว่าเจ้าของกุญแจ จะใช้กุญแจคู่นี้ในการติดต่อสื่อสารกับบุคคลอื่น โดยไม่จำเป็นต้องมีกุญแจเป็นจำนวนมาก

เช่นเดียวกับกรณีของระบบรหัสแบบสมมาตร ทั้งนี้ในการเก็บรักษากุญแจนั้น กุญแจส่วนตัวจะต้องเก็บรักษาไว้กับเจ้าของกุญแจและห้ามไม่ให้ผู้อื่นล่วงรู้ ส่วนกุญแจสาธารณะนั้นต้องมีการประกาศให้ผู้อื่นรับรู้หรือเก็บไว้ในที่ซึ่งบุคคลอื่นสามารถเข้ามาสืบค้นได้ ในการเข้ารหัสลับข้อมูลจะต้องใช้กุญแจดอกหนึ่ง ในการเข้ารหัสลับ และใช้กุญแจอีกดอกหนึ่งที่เป็นคู่กันในการถอดรหัสลับ เช่น หากใช้กุญแจสาธารณะในการเข้ารหัสลับก็จะต้องใช้กุญแจส่วนตัวในการถอดรหัสลับ ในทางกลับกัน หากใช้กุญแจส่วนตัวในการเข้ารหัสลับก็จะต้องใช้กุญแจสาธารณะในการถอดรหัสลับ ซึ่งวิธีการเลือกใช้กุญแจจะขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน เช่น นำไปใช้ในการเข้ารหัสลับ (Encryption) เพื่อรักษาความลับของข้อมูล หรือใช้งานในลักษณะลายมือชื่อดิจิทัล (Digital Signature) ขนาดของกุญแจที่นิยมใช้ในระบบรหัสแบบอสมมาตร คือ 512 – 2048 บิต ตัวอย่างวิธีการเข้า/ถอดรหัสลับของระบบรหัสแบบอสมมาตร เช่น RSA,DSA เป็นต้น



รูปที่ 2.2 การเข้า / ถอดรหัสแบบอสมมาตร

ข้อดีของระบบแบบอสมมาตร	ข้อเสียของระบบแบบอสมมาตร
- การบริหารกุญแจ (Key Management) สามารถทำได้ดีและเป็นระบบเนื่องจากผู้ใช้แต่ละคนต้องการกุญแจ 1 คู่ (กุญแจส่วนตัวและกุญแจสาธารณะ) เท่านั้น - สามารถใช้ในการรักษาความลับของข้อมูล ใช้ในการพิสูจน์ตัวจริง ใช้ในการตรวจสอบความครบถ้วนของข้อมูลและการปฏิเสธความรับผิด	- การเข้าและถอดรหัสจะช้ากว่า ระบบรหัสแบบสมมาตร ประมาณ 100 – 1000 เท่า

2.2 X.509 Certificate

มาตรฐาน X.509 Certificate ซึ่งเป็นมาตรฐานที่กำหนดโดย ITU-T X.509 และมีเริ่มประกาศใช้ในปี 1988 ปัจจุบันมาตรฐาน X.509 Certificate ได้รับการพัฒนาเป็นเวอร์ชัน 3 เพื่อปรับปรุงข้อบกพร่องของเวอร์ชันก่อนหน้า อีกทั้งยังมีการข้อกำหนดต่างๆ ในกรณีที่มีการเพิ่มข้อมูลในใบรับรองอิเล็กทรอนิกส์

โดยรูปแบบของใบรับรองนั้นประกอบด้วยข้อมูลหลัก 2 ส่วนคือ ข้อมูลพื้นฐานของใบรับรองอิเล็กทรอนิกส์ (Certificate Basic field) และข้อมูลเพิ่มเติมของใบรับรองอิเล็กทรอนิกส์ (Certificate Extension field)

รูปที่ 2.3 โครงสร้างของใบรับรองตามมาตรฐาน X.509

2.2.1 ข้อมูลพื้นฐานของใบรับรองอิเล็กทรอนิกส์ (Certificate Basic fields)

ข้อมูลพื้นฐานของใบรับรองอิเล็กทรอนิกส์นั้นมีมาตั้งแต่ X.509 เวอร์ชัน 1 ซึ่งจำเป็นต้องมีการกำหนดค่าลงไปในใบรับรองอิเล็กทรอนิกส์ ซึ่งประกอบด้วยข้อมูลที่จำเป็น ดังนี้

Version	เวอร์ชันของมาตรฐาน X.509 ที่ใบรับรองอิเล็กทรอนิกส์ใช้ ซึ่งปัจจุบันเวอร์ชัน 3 เป็นเวอร์ชันที่ผู้ให้บริการออกใบรับรอง (CA) ใช้กันทั่วโลก
Serial number	เป็นหมายเลขฐานสิบหกที่ CA กำหนดให้กับใบรับรองอิเล็กทรอนิกส์แต่ละใบ เพื่อป้องกันปัญหาการซ้ำกันของใบรับรอง โดยความยาวของ

	หมายเลขใบรับรองจะขึ้นอยู่กับ CA เป็นผู้กำหนด เช่น 0D44 3F, 0444 74FB, 198B 11D1 3F9A 8FFE 69A0
Signature algorithm	เป็นอัลกอริทึมที่ CA ใช้ในการลงลายมือชื่อดิจิทัลเพื่อรับรองใบรับรองอิเล็กทรอนิกส์ เช่น sha1WithRSAEncryption, md5WithDSAEncryption เป็นต้น จากตัวอย่าง sha1 และ md5 เป็นอัลกอริทึมในการย่อยข้อมูล (Hash function) ส่วน RSA และ DSA เป็นอัลกอริทึมในการสร้างลายมือชื่อดิจิทัล
Issuer	ชื่อของ CA ที่ทำการรับรองและออกใบรับรองอิเล็กทรอนิกส์ ซึ่งข้อมูลดังกล่าวจะอยู่ในรูปแบบ distinguished name (DN) ตามมาตรฐาน X.500 เช่น CN = G-CA Government Personnel CA,OU = G-CA Government Personnel CA,O = Government Information Technology Services,C = TH
Validity	ช่วงเวลาที่สามารถใช้งานใบรับรองอิเล็กทรอนิกส์ดังกล่าวได้ โดยมีการระบุวันที่เริ่มใช้งานและวันที่หมดอายุ ซึ่งมาตรฐานในการกำหนดวันที่มี 2 แบบ UTCTime มีรูปแบบคือ YYMMDDHHMMSSZ GeneralizedTime มีรูปแบบคือ YYYYMMDDHHMMSSZ UTCTime มีการกำหนดปีด้วยตัวเลข 2 ตัว คือ YY จะทำให้รูปแบบของใบรับรองอิเล็กทรอนิกส์ดังกล่าวใช้งานได้จนถึงปี 2049 เท่านั้น เนื่องจาก UTCTime เริ่มใช้เมื่อปี 1950 ซึ่งหากมีการใช้งานจนถึงปี 2050 จะทำให้ใบรับรองอิเล็กทรอนิกส์มีความหมายว่าเริ่มงานได้ในปี 1950 ดังนั้นในปัจจุบัน CA จึงใช้รูปแบบ GeneralizedTime เพื่อป้องกันปัญหาดังกล่าว
Subject	ชื่อของเจ้าของใบรับรองอิเล็กทรอนิกส์ ซึ่งอยู่ในรูปแบบ distinguished name (DN) ตามมาตรฐาน X.500 เหมือนกับ Issuer เช่น CN= Pativeth Ruangsuwan, OU=Faculty of Information Technology, O= King Mongkut's Institute of Technology Ladkrabang, C=TH

Subject Public key information	กุญแจสาธารณะ (Public Key) ของเจ้าของใบรับรองอิเล็กทรอนิกส์ และอัลกอริทึมที่ใช้ในการสร้างกุญแจสาธารณะ ตัวอย่างเช่น RSA, DSA, Diffie-Hellman เป็นต้น

2.2.2 ข้อมูลเพิ่มเติมของใบรับรองอิเล็กทรอนิกส์ (Certificate Extensions field)

ข้อมูลเพิ่มเติมของใบรับรองอิเล็กทรอนิกส์นั้นเป็นข้อมูลที่มีการกำหนดขึ้นมาเพิ่มเติมในเวอร์ชัน 3 เพื่อ โดยมีมีข้อมูลเพิ่มเติมนี้ประกอบด้วย 3 ส่วน คือ ชนิดของข้อมูลเพิ่มเติม (Type) ความจำเป็นของข้อมูล (Criticality) และค่าของข้อมูล (Value)

ชนิดของข้อมูล	ความจำเป็น	ข้อมูล

- ชนิดของข้อมูล (Type) เป็นข้อมูลที่มีการขึ้นทะเบียนเพื่อกำหนดหมายเลข ที่เรียกว่า Object ID เพื่อให้สามารถติดต่อสื่อสารกันได้อย่างเข้าใจ เช่น 2.5.29.3 = Certificate Policies, Key Usage = 2.5.29.15
- ความจำเป็นของข้อมูล (Criticality) เพื่อบอกแอพพลิเคชั่นว่าชนิดของข้อมูลนี้มีความจำเป็นต้องนำมาใช้ในการพิจารณาหรือไม่ เช่น Key Usage เป็นชนิดของข้อมูลที่มีค่าเป็น critical ซึ่งชนิดของข้อมูลนี้เป็นสิ่งที่บอกว่าใบรับรองอิเล็กทรอนิกส์นี้สามารถใช้งานในด้านใดได้บ้าง ดังนั้น แอพพลิเคชั่นจำเป็นต้องอ่านข้อมูลนี้เพื่อตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์นี้สามารถใช้งานกับแอพพลิเคชันดังกล่าวได้หรือไม่
- ค่าของข้อมูล (Value) คือค่าข้อมูลเพิ่มเติม

ข้อมูลเพิ่มเติมของใบรับรองอิเล็กทรอนิกส์นี้แบ่งเป็น 2 ประเภท คือ

2.2.2.1 Standard Extensions คือ ข้อมูลเพิ่มเติมที่เป็นมาตรฐานที่ใช้ในการติดต่อสื่อสาร ประกอบด้วยข้อมูลดังนี้

Authority Key Identifier	เป็นสิ่งที่ระบุถึงกุญแจสาธารณะ (Public Key) ที่เป็นคู่กับกุญแจส่วนตัว (Private Key) ที่ CA ใช้ในการลงลายมือชื่อดิจิทัลรับรองใบรับรองอิเล็กทรอนิกส์ เพื่อใช้ในการตรวจสอบว่าใบรับรองดังกล่าวออกโดย CA ใด

Subject Key Identifier	เป็นสิ่งที่ระบุถึง Public Key ของเจ้าของใบรับรองใบรับรอง ซึ่ง Subject Key Identifier นี้ จะใช้ร่วมกับ Authority Key Identifier ในการสร้าง Certification path เช่น ในการตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์ของผู้ใช้นั้นออกโดย CA รายใด จะดำเนินการโดยนำ Subject Key Identifier ของใบรับรองอิเล็กทรอนิกส์ของผู้ใช้ ไปตรวจสอบกับ Authority Key Identifier ของใบรับรองอิเล็กทรอนิกส์ของ CA ที่ต้องการตรวจสอบ หากตรงกันก็แสดงว่าใบรับรองอิเล็กทรอนิกส์ดังกล่าวเป็นของ CA ที่ออกใบรับรองอิเล็กทรอนิกส์ของผู้ใช้
Key Usage	เป็นข้อมูลที่บอกถึงวัตถุประสงค์ของกุญแจว่าสามารถนำไปใช้งานในด้านใด เช่น ใช้ในการลงลายมือชื่อดิจิทัล การเข้ารหัสลับข้อมูล การออกใบรับรองอิเล็กทรอนิกส์ เป็นต้น
Private Key Usage Period	เป็นข้อมูลที่ระบุระยะเวลาในการใช้งาน Private Key ที่คู่กับกุญแจส่วนตัวที่เป็นคู่กับกุญแจสาธารณะในใบรับรองอิเล็กทรอนิกส์
Certificate Policies	เป็นข้อมูลที่ระบุถึงนโยบายของผู้ให้บริการออกใบรับรองที่ใช้ในการออกใบรับรองอิเล็กทรอนิกส์ โดยข้อมูลนี้จะปรากฏทั้งในใบรับรองอิเล็กทรอนิกส์ของผู้ใช้และของ CA
Policy Mappings	เป็นข้อมูลที่ใช้ในกรณีที่มีการทำ Cross Certification ระหว่าง 2 CA โดยใช้ในการเปรียบเทียบ Certificate Policy ที่ 2 CA เข้ากันได้
Subject Alternative Name	เป็นข้อมูลเพิ่มเติมในการระบุชื่อใบรับรองอิเล็กทรอนิกส์ ในกรณีที่มีชื่อแตกต่างกันมากกว่า 1 ชื่อ เช่น e-mail address, Domain Name, IP Address, URI
Issuer Alternative Names	เป็นข้อมูลเพิ่มเติมในการระบุชื่อของ CA ที่ออกใบรับรองอิเล็กทรอนิกส์ ในกรณีที่มีชื่อแตกต่างกันมากกว่า 1 ชื่อ ซึ่งข้อมูลมีลักษณะเหมือนกับ Subject Alternative Name
Subject Directory	เป็นข้อมูลที่ระบุถึงข้อมูลเพิ่มเติมที่อยู่ในไดเรกทอรี X.500 ซึ่งเป็นส่วน

Attributes	หนึ่งในชื่อของใบรับรองอิเล็กทรอนิกส์
Basic Constraints	เป็นข้อมูลที่ใช้ในการระบุว่าใบรับรองอิเล็กทรอนิกส์ดังกล่าวเป็น ของ CA หรือไม่ โดยมีข้อมูลย่อย ดังนี้ - CA เป็นข้อมูลที่บ่งบอกว่าใบรับรองอิเล็กทรอนิกส์ ดังกล่าวเป็นของ CA หรือไม่ โดยหากมีค่าเป็น TRUE จะเป็นใบรับรองอิเล็กทรอนิกส์ของ CA หากมีค่าเป็น FALSE จะเป็นใบรับรองอิเล็กทรอนิกส์ของผู้ใช้
Path Length Constraints	เป็นข้อมูลที่บอกว่ามีใบรับรองอิเล็กทรอนิกส์ที่เป็น Intermediate ใน Certification Path
Name Constraints	เป็นข้อมูลที่ใช้ในการกำหนดขอบเขตของใบรับรองอิเล็กทรอนิกส์ ที่ออก เช่น หากใบรับรองอิเล็กทรอนิกส์ของ CA มีค่า NameConstraints= 'o=GITS' หมายความว่า ใบรับรองอิเล็กทรอนิกส์ที่ออกโดย CA นี้ จะต้องเป็นใบรับรองอิเล็กทรอนิกส์ที่มีชื่อองค์กร (Organization Name) เป็น GITS
Policy Constraints	เป็นข้อมูลที่ใช้ในการกำหนดขอบเขตการเชื่อถือใบรับรองอิเล็กทรอนิกส์ของ CA ที่ทำการ Cross Certification กัน เช่น CA1 จะรับรองใบรับรองอิเล็กทรอนิกส์ที่ออกโดย CA2 ก็ต่อเมื่อใบรับรองอิเล็กทรอนิกส์ใบนั้นจะต้องมี Policy OID ที่ปรากฏใน Policy Constraints ที่ตรงกับที่กำหนดไว้
Extended Key Usage	เป็นข้อมูลที่บอกถึงวัตถุประสงค์ของกุญแจที่เพิ่มเติมจากข้อมูลใน Key Usage ซึ่งอาจมีการกำหนดเพิ่มเติมขึ้นมาได้ แต่จำเป็นต้องมีการขึ้นทะเบียนหมายเลข OID จาก IANA หรือ ITU-T Recommendation X.660 [X.660]
CRL Distribution Points	เป็นข้อมูลที่บอกวิธีในการเข้าถึงรายการเพิกถอนใบรับรอง (CRL) ของ CA เพื่อให้แอพพลิเคชั่นสามารถนำข้อมูลนี้ไปใช้ตรวจสอบสถานะของใบรับรองอิเล็กทรอนิกส์ได้
Inhibit Any-Policy	เป็นข้อมูลที่ระบุถึงนโยบายพิเศษของใบรับรองอิเล็กทรอนิกส์ที่

	นอกเหนือจากนโยบายอื่นของใบรับรอง
Freshest CRL	เป็นข้อมูลระบุถึงวิธีการในการเข้าถึงรายการเพิกถอนใบรับรองย่อย (delta CRL) รายการล่าสุดของผู้ให้บริการออกใบรับรอง

2.2.2 Private Internet Extensions เป็น Extensions ที่กำหนดขึ้นมาเพื่อใช้งานเฉพาะทาง โดยปัจจุบันมี Extensions ในกลุ่มนี้ 2 Extension ด้วยกัน ดังนี้

Authority Information Access	เป็นข้อมูลที่ระบุวิธีการเข้าถึงบริการและนโยบายของ CA ที่ออกใบรับรองอิเล็กทรอนิกส์ที่มีข้อมูลนี้ ซึ่งบริการนี้รวมไปถึงบริการตรวจสอบสถานะของใบรับรองอิเล็กทรอนิกส์แบบออนไลน์ (OCSP) ด้วย
Subject Information Access	เป็นข้อมูลที่ระบุถึงวิธีการในการเข้าถึงข้อมูลและบริการ สำหรับใบรับรองอิเล็กทรอนิกส์ที่มีข้อมูลนี้ปรากฏอยู่

2.3 Time Stamp Protocol

2.3.1 ผู้ให้บริการประทับเวลา

ผู้ให้บริการประทับเวลา (Time Stamping Authority: TSA) เป็นหน่วยงานกลางที่เชื่อถือได้ทำหน้าที่สร้าง Time-Stamp (หรือ Time-Stamp Token) เพื่อจะบ่งบอกว่าข้อมูลได้ถูกสร้างขึ้นและมีอยู่ก่อนเวลาที่ได้มีการประทับเวลาไว้ ผู้ให้บริการประทับเวลาจะต้องมีคุณสมบัติเบื้องต้นดังนี้

1) ใช้แหล่งเวลาที่มีความน่าเชื่อถือ
2) ใส่ค่าของเวลาที่มีความน่าเชื่อถือลงในแต่ละ Time-Stamp Token
3) ใส่เลขจำนวนเต็มซึ่งไม่ซ้ำกันลงในแต่ละ Time-Stamp Token ที่สร้างขึ้นใหม่
4) สร้าง Time-Stamp Token เมื่อได้รับ Time-Stamp Request ที่ถูกต้องจากผู้ร้องขอ
5) ใส่ Security Policy ID ใน Time-Stamp Token ที่สร้างขึ้น
6) ประทับเวลาเฉพาะค่า Hash ของข้อมูลซึ่งผ่านฟังก์ชัน Hash ที่มีคุณสมบัติ Collision Resistant เท่านั้น

7) ตรวจสอบว่าความยาวของค่า Hash ที่ส่งมาต้องสอดคล้องกับความยาวของผลลัพธ์ของ Hash อัลกอริทึม
8) ไม่ตรวจสอบค่า Hash ที่จะประทับเวลาไม่ว่าทางใดก็ตาม นอกจากตรวจสอบความยาวของค่า Hash
9) ไม่ใส่สิ่งบ่งบอกผู้ขอประทับใดๆ ลงใน Time-Stamp Token
10) ลงลายมือชื่อแต่ละ Time-Stamp Token โดยใช้กุญแจซึ่งสร้างมาเพื่อการประทับเวลาโดยเฉพาะ และคุณสมบัติดังกล่าวถูกระบุในใบรับรองของกุญแจนั้น
11) ใส่ข้อมูลเพิ่มเติมลงใน Time-Stamp Token ตามที่ผู้ขอประทับเวลาร้องขอ โดยใช้ Extension Field เท่าที่รองรับโดยผู้ให้บริการประทับเวลานั้น หากไม่รองรับผู้ให้บริการประทับเวลาควรจะส่งข้อความผิดพลาดกลับไป

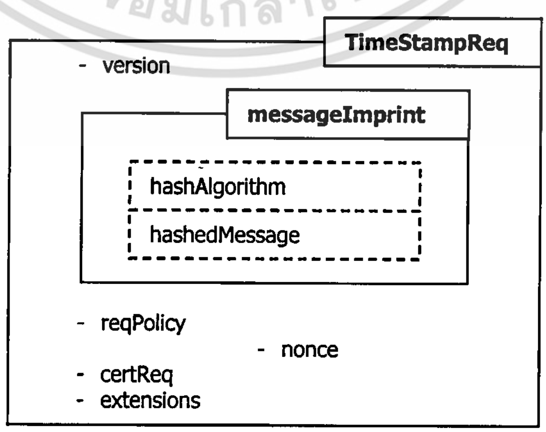
ผู้ให้บริการประทับเวลาจะต้องลงลายมือชื่อในการประทับเวลาแต่ละครั้งด้วยกุญแจซึ่งสงวนไว้สำหรับจุดประสงค์นั้นโดยเฉพาะ ผู้ให้บริการประทับเวลาหนึ่งๆ อาจมีกุญแจส่วนตัวได้หลายดอก เพื่อใช้สำหรับหลายนโยบาย หลายอัลกอริทึม หลายขนาดเพื่อเพิ่มสมรรถภาพ

ใบรับรองที่ใช้ในการประทับเวลาต้องเพียงมี 1 Extended Key Usage Field Extension [RFC 2459] และต้องมีค่าเท่ากับหมายเลข OID ระบุถึง id-kp-timeStamping

2.3.2 ขั้นตอนการประทับเวลา

ขั้นตอนการประทับเวลามีดังนี้

2.3.2.1 ผู้ขอประทับเวลาส่งข้อมูลที่ต้องการประทับเวลา ในรูปของ Time-Stamp Request ไปยังผู้ให้บริการประทับเวลา ซึ่งรูปแบบดังกล่าวสามารถแสดงในรูปที่ 2.4



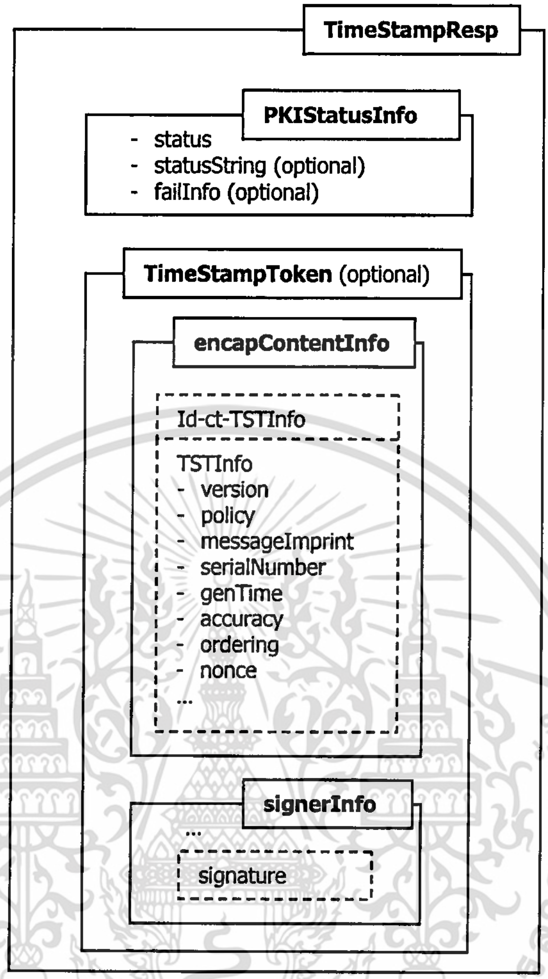
รูปที่ 2.4 รูปแบบของ Time-Stamp Request

ใน Time-Stamp Request ประกอบด้วยฟิลด์ต่อไปนี้

version	เวอร์ชั่นของของ Time-Stamp Request (ปัจจุบันเท่ากับ v1)
messageImprint	ซึ่งประกอบด้วย 2 ฟิลด์ ได้แก่ - hashAlgorithm ระบุถึงอัลกอริทึมในการ Hash ที่ใช้ในการสร้าง hashedMessage (ถ้าผู้ให้บริการประทับเวลาไม่รู้จักอัลกอริทึมดังกล่าว หรือทราบว่าอัลกอริทึมดังกล่าว ไม่มีคุณสมบัติ Collision Resistant ผู้ให้บริการประทับเวลาควรปฏิเสธที่จะส่ง Time-Stamp Token กลับ พร้อมทั้งระบุใน TimeStampResp.pkiStatusInfo.failInfo เท่ากับ 0 (badInfo) - hashedMessage ระบุค่า Message Digest ของข้อมูลที่ต้องการประทับเวลา
reqPolicy (Optional)	หมายเลข OID ที่ระบุ TSAPolicyId ที่ต้องการ
nonce (Optional)	เลขจำนวนเต็มขนาดใหญ่ สร้างโดยผู้ขอประทับเวลา เพื่อยันยืนความถูกต้องของเวลาใน ถ้าหากฟิลด์นี้มีค่าจะต้องตรงกับฟิลด์ nonce ใน Time-Stamp Response
certReq	หากมีค่า true ใบรับรองของผู้ให้บริการประทับควรถูกส่งกลับมาด้วยในฟิลด์ certificates ใน Time-Stamp Token หากมีค่า false ต้องไม่มีค่าในฟิลด์ certificates ใน Time-Stamp Token
extensions (Optional)	เพื่อรองรับข้อมูลการขอประทับเวลาเพิ่มเติมในอนาคต

Time-Stamp Request จะไม่ระบุถึงผู้ขอประทับเวลา ในสถานการณ์ที่ผู้ให้บริการประทับเวลาต้องการระบุถึงตัวตนของผู้ขอประทับเวลา ต้องมีวิถีทางในการระบุ / ตรวจสอบตัวตน เช่น CMS encapsulation หรือ TLS authentication

2.3.2.2 ผู้ให้บริการประทับเวลาทำการประทับเวลาข้อมูลที่ได้รับ แล้วส่งกลับในรูปของ Time-Stamp Response กลับไปยังผู้ขอประทับเวลา โดยมีรูปแบบตามรูปที่ 2.5



รูปที่ 2.5 รูปแบบของ Time-Stamp Response

จากรูปที่ 2.5 Time-Stamp Response ประกอบด้วย 2 ฟิลด์ได้แก่

1. PKIStatusInfo ระบุข้อมูลสถานะของการประทับเวลา ประกอบด้วย

status	เป็นหมายเลขแสดงสถานะของการประทับเวลา มีค่าดังตารางที่ 2.1 ถ้า status เป็น 0 หรือ 1 จะมี TimeStampToken นอกนั้นจะไม่มี TimeStampToken
statusString	ระบุข้อมูลสถานะของการประทับเวลาในลักษณะข้อความ
failInfo	ระบุถึงเหตุผลที่ Time-Stamp Request ถูกปฏิเสธ กรณีที่ไม่ส่ง TimeStampToken กลับมา ดังตารางที่ 2.2

ตารางที่ 2.1 สถานะของการประทับเวลา

ค่า	ความหมาย	คำอธิบาย
0	granted	อนุญาตให้มีการประทับเวลา
1	grantedwithMods	อนุญาตให้มีการประทับเวลาพร้อมกับการแก้ไข
2	rejection	ปฏิเสธการประทับเวลา
3	waiting	รอการประทับเวลา
4	revocationWarning	กำลังจะเกิดการเพิกถอนขึ้น
5	revocationNotification	เกิดการเพิกถอนขึ้นแล้ว

ตารางที่ 2.2 คำอธิบายถึงเหตุผลที่ไม่ส่ง TimeStampToken กลับมา

ค่า	ความหมาย	คำอธิบาย
0	badAlg	อัลกอริทึมไม่รู้จัก หรือไม่รองรับ (ไม่ปลอดภัย)
2	badRequest	การติดต่อสื่อสารไม่ได้รับอนุญาต หรือไม่รองรับ
5	badDataFormat	รูปแบบข้อมูลที่ส่งมาไม่ถูกต้อง
14	timeNotAvailable	ไม่มีแหล่งเวลาของผู้ให้บริการประทับเวลา
15	unacceptedPolicy	TSA policy ที่ระบุมาไม่รองรับ
16	unacceptedExtension	extension ที่ระบุมาไม่รองรับ
17	addInfoNotAvailable	ข้อมูลเพิ่มเติมที่ขอมาไม่มี หรือไม่เข้าใจ
25	systemFailure	ไม่สามารถจัดการกับการร้องขอได้ เนื่องจากระบบล้มเหลว

TimeStampToken ประกอบด้วยข้อมูลซึ่งเกิดจากการประทับเวลา มีชนิดเป็น signedData ทั้งนี้ ส่วนสำคัญที่บ่งบอกถึงเวลา ได้ระบุไว้ในฟิลด์ encapContentInfo.TSTInfo ประกอบด้วยฟิลด์ที่สำคัญได้แก่

version	ระบุถึงเวอร์ชันของ Time-Stamp Token (ปัจจุบันเท่ากับ v1)
policy	หมายเลข OID ซึ่งระบุถึง TSAPolicyId ที่กำกับ Time-Stamp Response ที่ออกไป ซึ่งหากฟิลด์ policy ใน Time-Stamp Request มีค่า ฟิลด์นี้จะต้องมีค่าเท่ากัน
messageImprint	ซึ่งควรจะต้องมีค่าเท่ากับฟิลด์ messageImprint ที่ระบุใน Time-Stamp Request

serialNumber	ระบุถึงหมายเลขที่ออกโดยผู้ให้บริการประทับเวลา ซึ่งแตกต่างกันสำหรับแต่ละ TimeStampToken ที่ออกโดยผู้ให้บริการประทับเวลาหนึ่ง ๆ
genTime	ระบุถึงเวลาซึ่ง TimeStampToken ถูกสร้างขึ้นโดยผู้ให้บริการประทับเวลา มีรูปแบบเป็น GeneralizedTime
accuracy	ค่าความแปรปรวนของเวลาที่ออกโดยผู้ให้บริการประทับเวลา ดังนั้นเวลาที่ Time-Stamp Token ถูกสร้างขึ้นจะเท่ากับ genTime ± accuracy
ordering	หากไม่มีหรือมีค่าเท่ากับ false เวลาที่ออกจะสามารถเรียงได้กัน ก็ต่อเมื่อความแตกต่างระหว่างค่าของฟิลด์ genTime ของ Time-Stamp Token อันแรกกับอันที่สองมากกว่าผลรวมของ accuracy ของทั้ง 2 อัน หากมีค่าเท่ากับ true ทุกๆ Time-Stamp Token จะเรียงกันตามฟิลด์ genTime ได้เสมอ โดยไม่ต้องสนใจ accuracy
nonce	ต้องมีค่าหากฟิลด์ nonce ใน Time-Stamp Request มีค่า และค่าดังกล่าวต้องเท่ากัน

2.3.3 ผู้ขอประทับเวลาจะต้องตรวจสอบสถานะของข้อมูลที่ได้รับ รวมทั้งความถูกต้องของ TimeStampToken และลายมือชื่อดิจิทัลของ TimeStampToken

ทั้งนี้ การประทับเวลาให้กับข้อมูลที่มีการลงลายมือชื่อดิจิทัล ค่าของ hashedMessage ในฟิลด์ messageImprint ของ Time-Stamp Request จะเป็นค่า Hash ของ signature ในฟิลด์ signerInfo ของลายมือชื่อดิจิทัลดังกล่าว อย่างไรก็ตาม กระบวนการนี้ควรจะต้องเกิดขึ้นภายหลังจากลายมือชื่อดิจิทัลถูกสร้างขึ้นไม่นาน (ไม่เกิน 2-3 ชั่วโมง)

ขั้นตอนในการตรวจสอบความถูกต้องของลายมือชื่อดิจิทัลที่ผ่านการประทับเวลามีดังนี้

1) ตรวจสอบความถูกต้องของ TimeStampToken จากลายมือชื่อดิจิทัลของผู้ให้บริการประทับเวลา
2) ตรวจสอบความเชื่อมโยงของ TimeStampToken ดังกล่าวกับลายมือชื่อดิจิทัลของข้อมูลนั้น
3) ตรวจสอบความถูกต้องของใบรับรองของผู้ลงลายมือชื่อดิจิทัล
4) ตรวจสอบว่าวัน/เวลาที่ระบุไว้ใน TimeStampToken ต้องอยู่ในช่วงเวลาที่ใบรับรองสามารถใช้งานได้
5) หากใบรับรองถูกเพิกถอน ต้องมีการตรวจสอบวัน/เวลาที่เพิกถอน ซึ่งควรเป็นเวลาภายหลังจากวัน/เวลาที่ระบุไว้ใน TimeStampToken

อย่างไรก็ตาม อาจเกิดปัญหาในกรณีที่ผู้ขอประทับรอข้อมูลที่จะได้รับจากผู้ให้บริการประทับเวลานานเกินไป (อาจเกิดเนื่องจากมีการจู่โจมให้เกิดความล่าช้า) แต่ก็จะขึ้นอยู่กับวิธีการในการรับส่ง และปัจจัยด้านสภาพแวดล้อมอื่นๆ

2.3.3 วิธีการรับ-ส่งข้อมูล

มาตรฐาน [RFC 3161] ไม่มีข้อบังคับใดๆ เกี่ยวกับวิธีในการรับ-ส่งข้อมูลสำหรับการประทับเวลา แต่ได้กล่าวถึงวิธีการซึ่งสามารถนำมาใช้ได้ มีดังนี้

2.3.3.1 อีเมล์

การรับ-ส่งข้อมูลสำหรับการประทับเวลาผ่านทางอีเมล์ จะใช้ 2 MIME object ดังต่อไปนี้

MIME Request object For E-mail
Content-Type: application/timestamp-query
Content-Transfer-Encoding: base64
<<the ASN.1 DER-encoded Time-Stamp message, base64-encoded>>

MIME Response object For E-mail
Content-Type: application/timestamp-reply
Content-Transfer-Encoding: base64
<<the ASN.1 DER-encoded Time-Stamp message, base64-encoded>>

ในทางปฏิบัติควรมีการเพิ่มพารามิเตอร์ "name" และ "filename" ลงไปด้วย การเพิ่ม "filename" ลงไปด้วยจะทำให้สามารถทราบได้ว่าเป็นการ query หรือ reply เมื่อมีการบันทึกลงไฟล์ ชื่อไฟล์ที่ใช้ไม่ควรเกิน 8 อักขระ และตามด้วยนามสกุล 3 อักขระ นามสกุลที่ใช้ควรเป็นไปตามตารางที่ 2.3 เพื่อให้สามารถบ่งบอก MIME Type ได้

ตารางที่ 2.3 นามสกุลของไฟล์ที่เหมาะสมสำหรับแต่ละ MIME Type

MIME Type	File Extension
application/timestamp-query	.TSQ
application/timestamp-reply	.TSR

2.3.3.2 File Based Protocol

เป็นวิธีการรับ-ส่งข้อมูลการประทับเวลาผ่านไฟล์ โดยใช้โปรโตคอลที่ใช้รับ-ส่งไฟล์ เช่น FTP ไฟล์ที่ประกอบด้วยข้อมูลการประทับเวลา จะต้องมีเพียงข้อมูลจากผู้ให้บริการประทับเวลาถูกเข้ารหัสแบบ DER เพียง 1 ข้อมูลเท่านั้น ต้องไม่มี Header หรือข้อมูลเพิ่มเติมใดๆ

Time-Stamp Request ควรจะอยู่ในไฟล์ 1 ไฟล์ และมีนามสกุลเป็น .tsq (เหมือนกับ Time-Stamp Query) Time-Stamp Response ควรจะอยู่ในไฟล์ 1 ไฟล์ และมีนามสกุลเป็น .tsr (เหมือนกับ Time-Stamp Reply)

2.3.3.3 Socket Based Protocol

เป็นวิธีการรับ-ส่งข้อมูลการประทับเวลาผ่านโปรโตคอลซึ่งอยู่บนพื้นฐานโปรโตคอล TCP วิธีการดังกล่าวเหมาะสำหรับผู้ที่สามารถสร้าง Transaction และสามารถรอรับผลได้ มีขั้นตอนดังนี้

- ผู้ให้บริการประทับเวลารอรับการติดต่อที่พอร์ต 318 (หรือพอร์ตอื่นที่กำหนดไว้อย่างเหมาะสม)
- ผู้ขอประทับเวลาเชื่อมต่อพอร์ตดังกล่าว และส่งข้อความร้องขอประทับเวลา
- ผู้ให้บริการประทับเวลาส่งกลับข้อความตอบสนอง และ/หรือหมายเลขอ้างอิงเพื่อไว้ใช้อ้างอิงถึงลำดับของข้อความตอบสนองดังกล่าว (ถ้ามีข้อความตอบสนองหลายข้อความต่อ 1 การร้องขอ ผู้ประทับเวลาจะต้องส่งหมายเลขอ้างอิงดังกล่าวกลับมาด้วย)

การรับ-ส่งข้อมูลกันนั้น จะรับ-ส่ง "direct TCP-based TSA message" ซึ่งประกอบด้วยฟิลด์ต่อไปนี้

length (32 บิต), flag (8 บิต), value

ฟิลด์ length ระบุถึงจำนวนออกเต็ทที่เหลือในข้อความนี้ (ซึ่งเท่ากับจำนวนออกเต็ทของฟิลด์ value บวก 1)

ฟิลด์ flag ระบุถึงชื่อข้อความ ซึ่งจะระบุด้วยว่าฟิลด์ value จะมีความหมายอย่างไร แสดงในตารางที่ 2.4

ตารางที่ 2.4 รายละเอียดของฟิลด์ flag

ชื่อข้อความ	flag	value	หมายเหตุ
tsaMsg	'00'H	ข้อความจาก TSA ที่เข้ารหัสแบบ DER	ข้อความจากผู้บริการประทับเวลา
pollRep	'01'H	หมายเลขอ้างอิง (32 บิต), time-to-check-back (32 บิต)	ตอบสนองว่า TSA ไม่พร้อมสำหรับการส่งข้อความ พร้อมกับส่งเวลาในการรอกลับไป
pollReq	'02'H	หมายเลขอ้างอิง (32 บิต)	ร้องขอการตอบสนองจาก TSA
negPollRep	'03'H	ไม่มีการตอบสนองอีกต่อไป	
partialMsgRep	'04'H	หมายเลขอ้างอิงถัดไป (32 บิต), time-to-check-back (32 บิต), ข้อความจาก TSA ที่เข้ารหัสแบบ DER	ตอบสนองเพียงบางส่วน และส่งหมายเลขอ้างอิงใหม่ พร้อมเวลาโดยประมาณ เพื่อรอรับส่วนถัดไป
finalMsgRep	'05'H	การตอบสนองครั้งสุดท้าย	
errorMsgRep	'06'H	ข้อความผิดพลาด	เช่น หมายเลขอ้างอิงที่ร้องขอไม่มี

time-to-check-back เป็นเลขจำนวนเต็ม 32 บิต หมายถึงเวลาในหน่วยวินาที ซึ่งระบุถึงช่วงเวลาที่น้อยที่สุดที่ไคลเอนท์ควรตรวจสอบสถานะอีกครั้งหนึ่ง เพื่อเป็นเวลาโดยประมาณให้ไคลเอนท์ส่ง pollReq ถัดไป

ลำดับของข้อความอาจเป็นดังนี้

1. ผู้ขอส่ง tsaMsg และรับ pollRep หรือ negPollRep หรือ partialMsgRep หรือ finalMsgRep ในการตอบสนอง
2. ผู้ขอส่ง pollReq และรับ negPollRep หรือ partialMsgRep หรือ finalMsgRep หรือ errorMsgRep ในการตอบสนอง

2.3.3.4 HTTP

การรับ-ส่งข้อมูลสำหรับการประทับเวลาผ่านทางโปรโตคอล HTTP จะใช้ 2 MIME object ดังต่อไปนี้

MIME Request object For HTTP or HTTPS
Content-Type: application/timestamp-query
<<the ASN.1 DER-encoded Time-Stamp Request message>>

MIME Response object For HTTP or HTTPS
Content-Type: application/timestamp-reply
<<the ASN.1 DER-encoded Time-Stamp Response message>>

ในกรณีที่เซิร์ฟเวอร์ได้รับ Request ที่ไม่ถูกต้อง เซิร์ฟเวอร์อาจส่งกลับเป็นความผิดพลาดของ HTTP

บทที่ 3
การวิเคราะห์และออกแบบระบบ

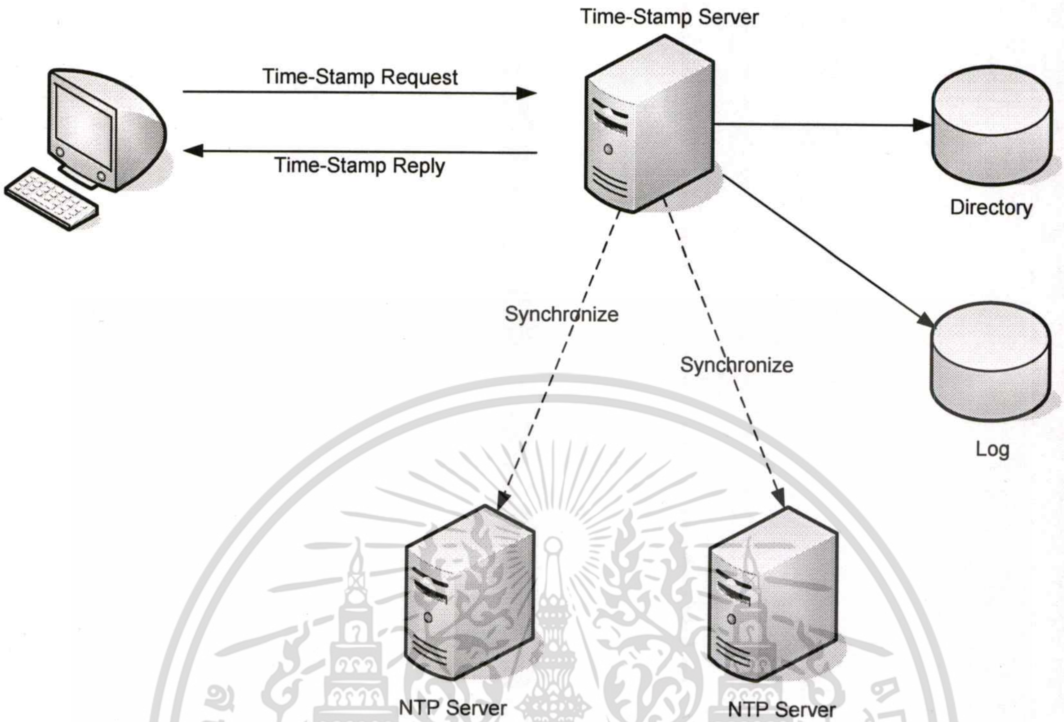
3.1 ปัญหาและแนวทางแก้ไข

ในองค์กรการทำงานต่าง ๆ ทุกกระบวนการทำงานล้วนต้องการความปลอดภัยในการดำเนินการไม่ว่าจะเป็นรูปแบบใด ๆ ก็ตามซึ่งในรูปแบบอิเล็กทรอนิกส์นั้นยิ่งต้องให้ความสำคัญในเรื่องความปลอดภัยไม่ว่าจะเป็นเรื่องของการรักษาความลับ (Confidentiality) ความครบถ้วนของข้อมูล (Data Integrity) การพิสูจน์ตัวตน (Authentication) รวมไปถึงการระบุเวลาที่เกิดขึ้นจริงเพื่อให้สามารถตรวจสอบผู้ส่ง-ผู้รับ และข้อมูลได้อย่างถูกต้อง

ในเรื่องของการประทับเวลาที่เกิดขึ้นจริงมีใช้กันอย่างแพร่หลายในการตรวจสอบความถูกต้องของเวลาในเอกสารหรือทรานแซกชันใด ๆ ที่เกิดขึ้นซึ่งเดิมในการประทับเวลาจะใช้เวลาที่เป็นเวลาของเครื่องคอมพิวเตอร์ของผู้ส่ง-ผู้รับ ซึ่งอาจเกิดปัญหาในเรื่องของความไม่เที่ยงตรงของเวลา การปลอมแปลงเวลา ทำให้ไม่สามารถตรวจสอบความถูกต้องของเวลาได้ จึงได้มีการนำกระบวนการประทับเวลาช่วยแก้ปัญหาที่เกิดขึ้นนี้

ระบบผู้ประทับเวลาเป็นระบบที่ให้บริการประทับเวลากับเอกสารหรือทรานแซคชันที่เกิดขึ้นเพื่อให้มีเวลาที่มีความเที่ยงตรงและไม่สามารถแก้ไขค่าของเวลาที่ออกให้ได้ โดยการประทับเวลาจะเริ่มขึ้นเมื่อผู้ขอประทับเวลาทำการร้องขอเพื่อประทับเวลาโดยทำการส่ง Time-Stamp Request ไปยังผู้ให้บริการประทับเวลา (Time-Stamp Authority) เมื่อผู้ให้บริการประทับเวลาตรวจสอบความถูกต้องของการร้องขอเรียบร้อย จะทำการสร้าง Time-Stamp Token และทำการลงลายมือชื่อดิจิทัลจึงทำการส่งกลับไปยังผู้ขอประทับเวลา โดยสามารถแสดงสถาปัตยกรรมของระบบได้ดังรูปที่ 3.1

3.2 สถาปัตยกรรมของระบบ (System Architecture)



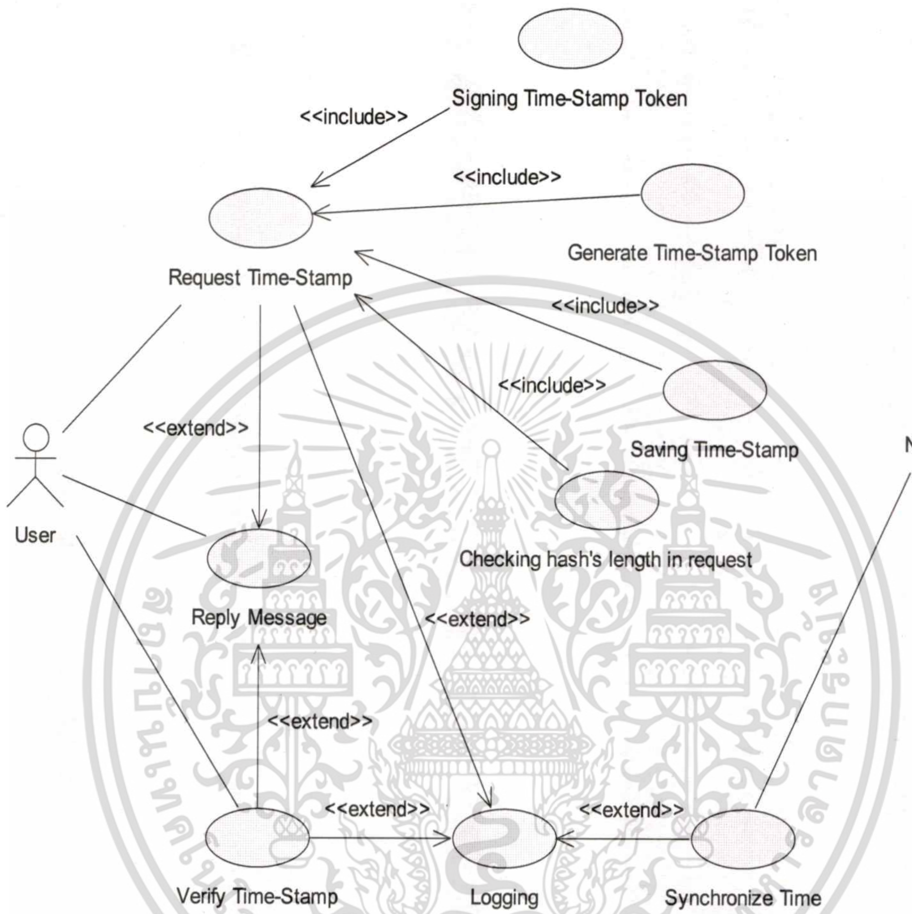
รูปที่ 3.1 สถาปัตยกรรมของระบบผู้ให้บริการประทับเวลา

สถาปัตยกรรมของระบบผู้ให้บริการประทับเวลา อธิบายได้ดังนี้

- ผู้ใช้งานทำการส่ง Request ผ่านทางเว็บแอพพลิเคชัน FTP,Socket หรืออีเมล์แอดเดรส
- Time-Stamp Server จะทำการตรวจสอบความถูกต้องของ Request และทำการสร้าง Time-Stamp และทำการส่งกลับไปยังผู้ใช้งาน
- Time-Stamp Server จะทำการบันทึก Time-Stamp ที่เกิดขึ้นลงในไดเรกทรอรีของระบบ
- Time-Stamp Server ทำการ Synchronize เวลากับ NTP Server
- เมื่อเกิด Activity ใด ๆ ในระบบ Time-Stamp Server จะทำการบันทึก Log ทุกครั้ง

3.3 ยูสเคสไดอะแกรม (Use Case Diagram)

ระบบผู้ให้บริการประทับเวลา สามารถเขียนเป็นยูสเคสเพื่อแสดงแอกเตอร์และรายละเอียดโมดูลหลัก ๆ ของระบบได้ดังนี้



รูปที่ 3.2 ยูสเคสไดอะแกรมของระบบผู้ให้บริการประทับเวลา

ยูสเคสไดอะแกรม จะมีแอกเตอร์ที่เป็นการแสดงถึงบุคคลที่เกี่ยวข้องกับระบบผู้ให้บริการประทับเวลา ซึ่งจากยูสเคสไดอะแกรมข้างต้น ประกอบด้วยแอกเตอร์ 2 แอกเตอร์ ดังต่อไปนี้

- **User** คือ ผู้ใช้งานระบบที่ต้องการร้องขอ time-stamp หรือต้องการตรวจสอบ time-stamp
- **NTP Server** คือผู้ที่ให้บริการ Time ให้กับระบบ โดยระบบจะทำการ Synchronize กับ NTP Server

สำหรับหน้าที่และการทำงานหลักของระบบ จากยูสเคสไดอะแกรมข้างต้นจะเห็นว่าประกอบด้วย ยูสเคส 9 ยูสเคส ดังต่อไปนี้

- **Request Time-Stamp** คือ การร้องรอ Time-Stamp token กับระบบ โดยจะมีการสร้าง Time-Stamp request ด้วย
- **Checking hash's length in request** คือ การตรวจสอบความถูกต้องของ Time-Stamp request ว่ามีความยาวตามอัลกอลิทึมที่ผู้ให้บริการอนุญาตหรือไม่
- **Generate Time-Stamp Token** คือ กระบวนการสร้าง Time-Stamp token เมื่อได้รับ Time-Stamp request ที่ถูกต้อง
- **Signing Time-Stamp Token** คือ การลงลายมือชื่อดิจิทัลให้กับ Time-Stamp token ซึ่งจะใช้อัลกอลิทึม SHA1WithRSAEncryption
- **Saving Time-Stamp** คือ การบันทึก Time-Stamp token ลงฐานข้อมูล
- **Verify Time-Stamp** คือ การตรวจสอบความถูกต้องของ Time-Stamp token ว่าตรงกับข้อมูลต้นฉบับหรือไม่ และตรวจสอบว่า Time-Stamp token นี้ถูกออกโดยผู้ให้บริการที่น่าเชื่อถือหรือไม่
- **Logging** คือ การบันทึกเหตุการณ์ต่าง ๆ ที่เกิดในระบบลง log เพื่อเป็นหลักฐานที่ใช้ในการตรวจสอบต่อไป
- **Synchronize Time** คือ การเข้าจังหวะเวลาเพื่อให้ระบบมีเวลาคลาดเคลื่อนน้อยที่สุด
- **Reply Message** คือ ระบบจะทำการส่งข้อความกลับไปยังผู้ร้องขอประทับเวลา หรือผู้ที่ทำการตรวจสอบความถูกต้องของข้อมูลประทับเวลา ไม่ว่าจะเป็นกรณีที่ขอประทับเวลาได้หรือไม่ได้ และตรวจสอบได้หรือไม่ได้เช่นกัน

หลังจากแสดงส่วนประกอบของยูสเคสไดอะแกรมทั้งแอกเตอร์และยูสเคส ตามที่ได้อธิบายไว้ข้างต้นแล้ว สามารถอธิบายรายละเอียดของขั้นตอนการทำงานแต่ละยูสเคสตามรายละเอียดในตารางที่ 3.1 ถึง ตารางที่ 3.9 ดังต่อไปนี้

ตารางที่ 3.1 รายละเอียดยูสเคส Request Time-Stamp

Use Case Name:	Request Time-Stamp	
Scenario:	Request Time-Stamp	
Triggering Event:	ผู้ใช้งานทำการร้องขอ Time-Stamp	
Brief Description:	เป็นส่วนที่รับการร้องขอ Time-Stamp จากผู้ใช้ และนำข้อมูลที่ผู้ใช้ส่งมาไปประมวลผลเพื่อสร้าง Time-Stamp	
Actors:	User	
Related Use Cases:	Include : Checking hash's length in request, Generate Time-Stamp Token, Signing Time-Stamp Token, Saving Time-Stamp Extend : Reply Message, Logging	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
	1. ผู้ใช้ทำการร้องขอ Time-Stamp และทำการส่ง Request มายัง Time-Stamp Server	1. ระบบทำการรับ Request 2. ระบบทำการตรวจสอบความถูกต้องของ Request 3. ระบบทำการสร้าง Time-Stamp 4. ระบบทำการลงลายมือชื่อดิจิทัลลงบน Time-Stamp ที่ได้สร้างขึ้น 5. ระบบทำการบันทึก Time-Stamp ลงบนไดเรกทรอรี 6. ระบบทำการส่ง Time-Stamp กลับไปยังผู้ที่ทำการร้องขอ 7. ระบบทำการบันทึก Log
Exception Conditions:	-	

ตารางที่ 3.2 รายละเอียดยูสเคส Checking hash's length in request

Use Case Name:	Checking hash's length in request	
Scenario:	Checking hash's length in request	
Triggering Event:	เมื่อระบบได้รับ Request เรียบร้อยแล้ว	
Brief Description:	เป็นส่วนที่ใช้สำหรับตรวจสอบความถูกต้องของ Request	
Actors:	-	
Related Use Cases:	Include : Request Time-Stamp	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
		1. ระบบทำการตรวจสอบความถูกต้องของ Request 2. ถ้าถูกต้องจะทำการ Generate Time-Stamp Token 3. แต่ถ้า Request ไม่ถูกต้อง ระบบจะทำการส่ง Reply Message กลับไปยังผู้ที่ทำการร้องขอ
Exception Conditions:	-	

ตารางที่ 3.3 รายละเอียดยูสเคส Generate Time-Stamp Token

Use Case Name:	Generate Time-Stamp Token
Scenario:	Generate Time-Stamp Token
Triggering Event:	เมื่อระบบทำการตรวจสอบว่า Request ถูกต้อง
Brief Description:	เป็นส่วนที่ใช้ในการสร้าง Time-Stamp
Actors:	-
Related Use Cases:	Include : Request Time-Stamp
Stakeholders:	-
Preconditions:	-
Postconditions:	-

ตารางที่ 3.3 (ต่อ)

Flow of Events:	**Actor**	**System**
		1. ระบบทำการตรวจสอบความถูกต้องของ Request 2. ถ้าถูกต้องจะทำการ Generate Time-Stamp Token 3. แต่ถ้า Request ไม่ถูกต้อง ระบบจะทำการส่ง Reply Message กลับไปยังผู้ที่ทำการร้องขอ
Exception Conditions:	-	

ตารางที่ 3.4 รายละเอียดยูสเคส Signing Time-Stamp Token

Use Case Name:	Signing Time-Stamp Token	
Scenario:	Signing Time-Stamp Token	
Triggering Event:	เมื่อระบบทำการสร้าง Time-Stamp เรียบร้อยแล้ว	
Brief Description:	เป็นส่วนที่ใช้ในลงลายมือชื่อดิจิทัลว่า Time-Stamp ที่ได้สร้างขึ้นมานี้เป็นของผู้ให้บริการจริง ๆ	
Actors:	-	
Related Use Cases:	Include : Request Time-Stamp	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
		1. ระบบทำการลงลายมือชื่อดิจิทัลโดยใช้ Private Key ของผู้ให้บริการ
Exception Conditions:	-	

ตารางที่ 3.5 รายละเอียดยูสเคส Saving Time-Stamp

Use Case Name:	Saving Time-Stamp	
Scenario:	Saving Time-Stamp	
Triggering Event:	เมื่อระบบทำการลงลายมือชื่อดิจิทัลเรียบร้อยแล้ว	
Brief Description:	เป็นส่วนที่ใช้บันทึก Time-Stamp ที่ได้สร้างขึ้น	
Actors:	-	
Related Use Cases:	Include : Request Time-Stamp	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
		1. ระบบทำการบันทึก Time-Stamp ที่ได้สร้างลงในไดเรกทอรี
Exception Conditions:	-	

ตารางที่ 3.6 รายละเอียดยูสเคส Synchronize Time

Use Case Name:	Synchronize Time	
Scenario:	Synchronize Time	
Triggering Event:		
Brief Description:	เป็นส่วน Synchronize time เพื่อให้ time ที่ใช้มีความถูกต้อง น่าเชื่อถือ	
Actors:	NTP Server	
Related Use Cases:	Extend : Logging	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
	1. NTP ทำการ Synchronize time กับระบบ	1. ระบบทำการ Set Time ให้ถูกต้อง 2. ระบบทำการบันทึก Log เพื่อใช้ในการตรวจสอบ
Exception Conditions:	-	

ตารางที่ 3.7 Verify Time-Stamp

Use Case Name:	Verify Time-Stamp	
Scenario:	Verify Time-Stamp	
Triggering Event:	User ทำการตรวจสอบ Time-Stamp ที่ออกโดยผู้ให้บริการ	
Brief Description:	User ทำการตรวจสอบ Time-Stamp ที่ออกโดยผู้ให้บริการ	
Actors:	User	
Related Use Cases:	Extend : Logging, Reply Message	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
	1. User ทำการส่ง Time-Stamp เพื่อตรวจสอบความถูกต้อง	1. ระบบทำการตรวจสอบ Time-Stamp 2. ระบบทำการส่ง Message ผลการตรวจสอบกลับไปยัง User 3. ระบบทำการบันทึก Log
Exception Conditions:	-	

ตารางที่ 3.8 รายละเอียดยูสเคส Reply Message

Use Case Name:	Reply Message
Scenario:	Reply Message
Triggering Event:	ระบบทำการตอบกลับไปยังผู้ใช้
Brief Description:	ระบบทำการตอบกลับไปยังผู้ใช้ในกรณีที่ Request ไม่ถูกต้อง,ส่ง Time-Stamp,ส่งผลการ Verify Time-Stamp
Actors:	
Related Use Cases:	Extend : Request Time-Stamp,Verify Time-Stamp
Stakeholders:	-
Preconditions:	-
Postconditions:	-

ตารางที่ 3.8 (ต่อ)

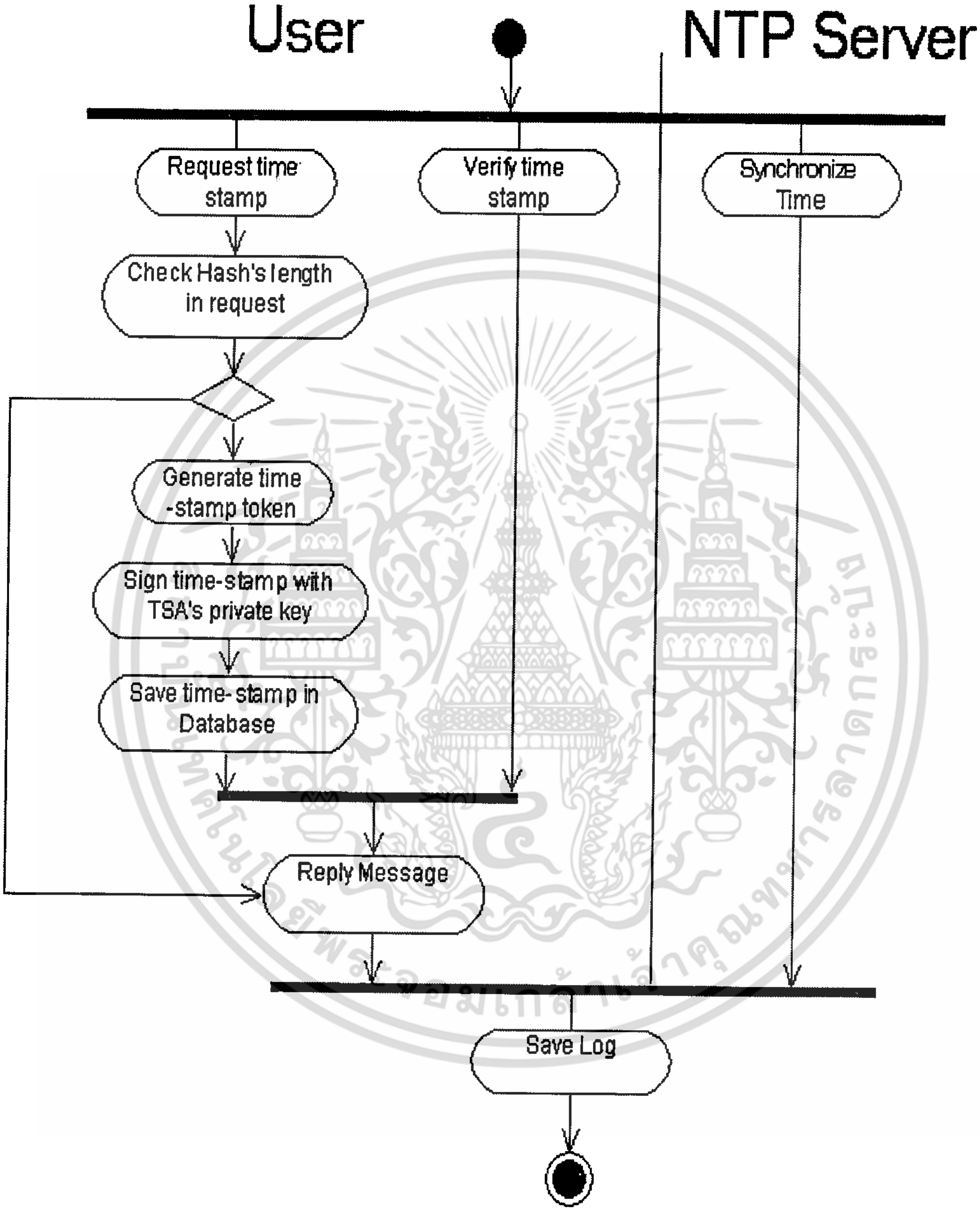
Flow of Events:	**Actor**	**System**
		1. ระบบทำการตรวจสอบ Time-Stamp 2. ระบบทำการส่ง Message ผลการตรวจสอบกลับไปยัง User 3. ระบบทำการบันทึก Log
Exception Conditions:	-	

ตารางที่ 3.9 รายละเอียดยูสเคส Logging

Use Case Name:	Logging	
Scenario:	Logging	
Triggering Event:	ระบบทำการบันทึก Log เมื่อเกิด Activity ขึ้น	
Brief Description:	ระบบทำการบันทึก Log เมื่อเกิด Activity ขึ้น ในทุก ๆ กรณี ไม่ว่าจะเกิดจาก User , Server เพื่อใช้ในการตรวจสอบ	
Actors:		
Related Use Cases:	Extend : Request Time-Stamp,Verify Time-Stamp,Synchronize Time	
Stakeholders:	-	
Preconditions:	-	
Postconditions:	-	
Flow of Events:	**Actor**	**System**
		1. ระบบทำการบันทึก Log ของ Activity ที่เกิดขึ้น
Exception Conditions:	-	

3.4 แอกทิวิตี้ไดอะแกรม (Activity Diagram)

แอกทิวิตี้ไดอะแกรมของระบบผู้ให้บริการประทับเวลา แสดงขั้นตอนในการร้องขอ Time-Stamp, การ Verify Time-Stamp และการ Synchronize Time ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 แอกทิวิตี้ไดอะแกรมแสดงการทำงานของระบบผู้ให้บริการประทับเวลา

สำหรับรายละเอียดขั้นตอนในการร้องขอ Time-Stamp, การ Verify Time-Stamp และการ Synchronize Time สามารถอธิบายได้ดังนี้

การสร้าง time-stamp

- เมื่อได้รับ time-stamp request ซึ่งจะอยู่ในรูปแบบ ASN.1 ระบบจะทำการตรวจสอบค่า messageImprint เพื่อดูว่ามีความยาวตรงตามอัลกอลิทึมที่กำหนดไว้หรือไม่ โดยค่า messageImprint จะมีรูปแบบเป็นเลขฐานสิบหก โดยขนาดความยาวของ messageImprint จะขึ้นอยู่กับอัลกอลิทึมที่ใช้สร้าง ซึ่งในโครงการนี้จะรองรับอัลกอลิทึม SHA-1 และ MD5 เท่านั้น ถ้า time-stamp request ที่ส่งมามีขนาดความยาวที่ไม่ได้เกิดจากการสร้างด้วยอัลกอลิทึม SHA-1 หรือ MD5 จะทำการปฏิเสธการสร้าง time-stamp token
- หลังจากที่ทำการตรวจสอบความยาวของ messageImprint ว่าถูกต้องแล้วจะทำการสร้าง time-stamp โดยจะดูค่าใน time-stamp request ดังต่อไปนี้
  - Nonce ว่ามีการส่งค่ามาหรือไม่ ถ้ามีการส่งมาจะนำค่า nonce นี้อยู่ใน time-stamp request ไปใส่ลงใน time-stamp token ที่ได้ทำการสร้างขึ้น
  - ReqPolicy ทำการตรวจสอบว่ามีการร้องขอ Policy ID ที่ถูกต้องหรือไม่ ถ้าไม่ถูกต้อง หรือไม่มี Policy ID ที่ได้ทำการร้องขอจะทำการปฏิเสธการสร้าง time-stamp token
  - ReqCert ตรวจสอบว่ามีการร้องขอ Certificate หรือไม่ถ้าทำการร้องขอจะทำการใส่ Certificate ลงใน time-stamp token ด้วย
- เมื่อทำการตรวจสอบค่าใน time-stamp request เสร็จจะทำการสร้าง time-stamp token
- เมื่อทำการสร้าง time-stamp token เสร็จจะทำการลงลายมือชื่อดิจิทัลด้วย private key ของผู้ให้บริการ ซึ่งการลงลายมือชื่อดิจิทัลให้กับ time-stamp token จะให้อัลกอลิทึม SHA1WithRSAEncryption คือ ทำการ hash time-stamp token ด้วย SHA-1 และทำการ Encrypt ด้วยอัลกอลิทึม RSA
- เมื่อทำการสร้าง time-stamp token เสร็จแล้ว จะทำการบันทึกค่า time-stamp token ลงในฐานข้อมูล
- เมื่อบันทึกเสร็จจะทำการส่ง time-stamp token กลับไปยังผู้ร้องขอ โดย ผู้ร้องรอจะต้องทำการตรวจสอบ time-stamp token ว่าถูกต้องหรือไม่ โดยตรวจสอบดังนี้
  - ตรวจสอบค่า nonce ว่าตรงกับค่า nonce ที่มีอยู่ใน time-stamp request หรือไม่
  - ตรวจสอบ Policy ID ว่าถูกต้องหรือไม่ ถ้ามีการร้องขอไป
  - ตรวจสอบ messageImprint ว่าตรงกับค่า messageImprint ใน time-stamp request หรือไม่
  - ตรวจสอบ Certificate ที่อยู่ใน time-stamp token ว่าเป็น Certificate ที่น่าเชื่อถือหรือไม่

การตรวจสอบ time-stamp

- ทำการเลือก time-stamp token และข้อมูลต้นฉบับ โดยจะตรวจสอบค่า hash ของข้อมูลต้นฉบับเปรียบเทียบกับค่าของ messageImprint ใน time-stamp token ถ้ามีค่าตรงกันแสดงว่าข้อมูลไม่มีการเปลี่ยนแปลงนับตั้งแต่นำข้อมูลไปทำ time-stamp แต่ถ้าค่าไม่ตรงกันแสดงว่ามีการเปลี่ยนแปลงหลังจากที่ได้มีการทำ time-stamp
- ตรวจสอบว่าลายมือชื่อดิจิทัลที่อยู่ใน time-stamp token นั้นว่าถูกสร้างจากใบรับรองที่ได้รับความไว้วางใจ (trust) หรือไม่และใบรับรองดังกล่าวยังอยู่ในสถานะที่ใช้งานได้หรือไม่

การ Synchronize time

- ทำการ Synchronize time กับ NTP Server ที่ให้บริการ Network time โดยจะทำการ synchronize กับ NTP Server ทุก ๆ 30 นาที

3.5 พจนานุกรมฐานข้อมูล (Data Dictionary)

ข้อมูลในระบบผู้ให้บริการประทับเวลา นำเสนอผ่านพจนานุกรมข้อมูลที่จะแสดงให้เห็นรายละเอียดของข้อมูลต่างๆ ที่เกี่ยวข้องกับการทำงานของระบบ ได้ดังตารางที่ 3.10

ตารางที่3.10 รายละเอียดข้อมูลตาราง Time-Stamp

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาวข้อมูล	คีย์	ตารางอ้างอิง
token_id	หมายเลขของ Time-Stamp token ซึ่งไม่ซ้ำกัน	Varchar	10	PK	
token_date	วัน-เวลาที่ระบุใน Time-Stamp token	Date/Time			
token_pkcs7	ค่า Time-Stamp token ซึ่งเก็บโดยใช้รูปแบบการเข้ารหัสแบบ DER	Blob			

3.6 การวิเคราะห์และออกแบบรับส่งข้อมูล

การร้องขอ time-stamp จะทำการร้องขอผ่าน HTTP Protocol โดยจะทำการร้องขอมาที่ http://tsa.proj.dev/tsa/ ซึ่งการร้องขอจะต้องทำการสร้าง time-stamp request ซึ่งมีโครงสร้างดังนี้

MIME Request object For HTTP or HTTPS
Content-Type: application/timestamp-query
<<the ASN.1 DER-encoded Time-Stamp Request message>>

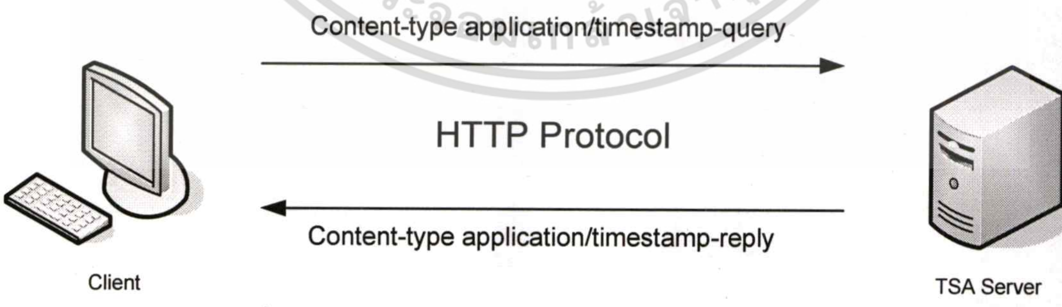
รูปที่ 3.4 โครงสร้าง time-stamp request

ข้อมูลจะมีรูปแบบมาตรฐาน ASN.1 โดยจะเก็บในรูปแบบไบนารี การส่ง time-stamp request จะมี Header เป็น Content-type application/timestamp-query

MIME Response object For HTTP or HTTPS
Content-Type: application/timestamp-reply
<<the ASN.1 DER-encoded Time-Stamp Response message>>

รูปที่ 3.5 โครงสร้าง time-stamp reply

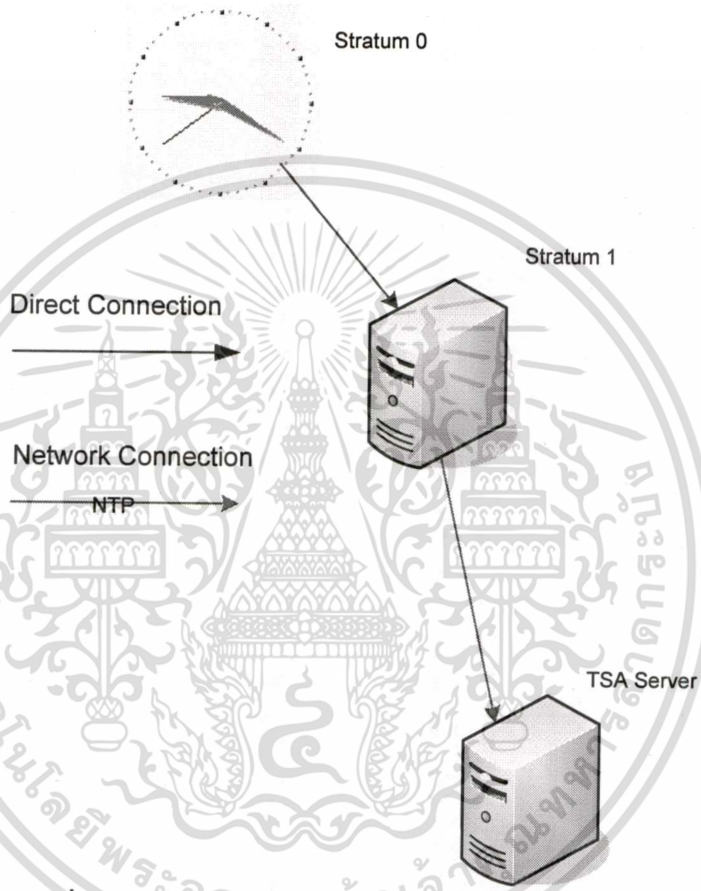
ข้อมูลจะมีรูปแบบมาตรฐาน ASN.1 โดยจะเก็บในรูปแบบไบนารี การส่ง time-stamp reply จะมี Header เป็น Content-type application/timestamp-reply
ซึ่งการรับส่งข้อมูลจะแสดงได้ตามรูปที่ 3.6



รูปที่ 3.6 การรับส่งข้อมูลของระบบผู้ให้บริการประทับเวลา

3.7 การ Synchronize Time

การ Synchronize time จะทำการ synchronize ทุกๆ 10 นาทีเพื่อให้เวลาที่ใช้ในการสร้าง time-stamp token มีความคลาดเคลื่อนน้อยที่สุด โดยจะทำการ synchronize กับ time server ntp.nasa.gov ซึ่งเป็น time server ใน stratum ที่ 1 ซึ่งการ synchronize time จะทำตาม RFC 1305 (Network Time Protocol) การ Synchronize time แสดงได้ดังรูป 3.7



รูปที่ 3.7 การ Synchronize time กับ NTP Server Stratum 1

3.8 การวิเคราะห์และออกแบบใบรับรองอิเล็กทรอนิกส์สำหรับผู้ให้บริการประทับเวลา

รายละเอียดของการวิเคราะห์และออกแบบใบรับรองอิเล็กทรอนิกส์สำหรับผู้ให้บริการประทับเวลา เป็นดังนี้

3.6.1 การวิเคราะห์โครงสร้างใบรับรองอิเล็กทรอนิกส์สำหรับผู้ให้บริการประทับเวลา

ข้อมูลในใบรับรองอิเล็กทรอนิกส์แบ่งออกเป็น 2 ส่วนคือ

1.ข้อมูลพื้นฐานของใบรับรองอิเล็กทรอนิกส์ (Certificate Basic Field)

2.ข้อมูลเพิ่มเติมของใบรับรองอิเล็กทรอนิกส์ (Certificate Extension Field)

ตารางที่ 3.11 รายละเอียด Certificate Basic Field

Certificate Basic Field	
Version	Version 3 เป็นเวอร์ชั่นที่ใช้ในปัจจุบันตามมาตรฐาน RFC 3828
Serial Number	เป็นหมายเลขที่ทางผู้ให้บริการใบรับรองอิเล็กทรอนิกส์เป็นผู้กำหนดให้ โดยจะเป็นเลขฐานสิบหก
Signature Algorithm	เป็นอัลกอริทึมที่ใช้ในการลงลายมือชื่อดิจิทัลให้กับใบรับรองอิเล็กทรอนิกส์
Issuer	เป็นชื่อของผู้ที่ออกใบรับรองอิเล็กทรอนิกส์ โดยจะมีรูปแบบตาม X.500
Validity	เป็นช่วงเวลาที่ใบรับรองอิเล็กทรอนิกส์สามารถใช้งานได้ โดยจะทำการเก็บเป็นรูปแบบUTC โดยอายุของใบรับรองจะขึ้นอยู่กับนโยบายของผู้ให้บริการว่าจะมีอายุเท่าไหร่
Subject	เป็นชื่อของเจ้าของใบรับรอง โดยจะมีรูปแบบตาม X.500
Subject Public Key Information	เป็นข้อมูลกุญแจสาธารณะที่เจ้าของกุญแจเป็นผู้สร้างเอง ซึ่งจะต้องมีความยาวอย่างน้อย 1024 bits เป็นอย่างต่ำ

ตารางที่ 3.12 รายละเอียด Certificate Extension Field

Certificate Extension Field	
Authority Key Identifier	มีการระบุค่าลงในฟิลด์ authorityKeyIdentifier เพื่อใช้ตรวจสอบว่าผู้ให้บริการออกใบรับรอง (CA) ใดเป็นผู้ออกใบรับรองดังกล่าว
Subject Key Identifier	มีการระบุค่าฟิลด์ subjectKeyIdentifier เพื่อเป็นตัวแทนของกุญแจสาธารณะที่อยู่ในใบรับรอง รวมทั้งยังใช้ในกระบวนการสร้าง Certification Path ดังนั้นจึงการกำหนดค่าลงในฟิลด์นี้
Key Usage	ระบุถึงการใช้งานของใบรับรองอิเล็กทรอนิกส์เพื่อให้แอปพริเคชั่นสามารถรู้ได้ว่าใบรับรองอิเล็กทรอนิกส์นี้สามารถใช้งานด้านใดได้บ้าง ใบรับรองอิเล็กทรอนิกส์สำหรับ Time-Stamp ควรจะมีเฉพาะ Digital Signature
Private Key Usage Period	ไม่มีการกำหนด เนื่องจากอายุใบรับรองที่ใช้สำหรับ Time-Stamp จะถูกกำหนดโดยผู้ให้บริการใบรับรองอิเล็กทรอนิกส์
Certificate Policy	มีการระบุนโยบายของผู้ให้บริการใบรับรองอิเล็กทรอนิกส์ เพื่อให้ใบรับรองอิเล็กทรอนิกส์ที่ใช้สำหรับ Time-Stamp มีความน่าเชื่อถือ
Policy Mappings	ไม่มีการกำหนดค่า เนื่องจากตามมาตรฐาน RFC 3280 ระบุว่าฟิลด์ policyMappings ไม่มีการใช้งานใน End Entity Certificate
Subject Alternative Name	ไม่มีการระบุค่าลงไป เนื่องจากใบรับรองอิเล็กทรอนิกส์สำหรับ Time-Stamp ไม่ได้ระบุชื่ออื่น ๆ ลงในใบรับรอง
Issuer Alternative Name	ขึ้นอยู่กับผู้ให้บริการใบรับรองอิเล็กทรอนิกส์ว่ามีชื่ออื่น ๆ ที่จะทำการใส่ลงไปหรือไม่
Subject DirectoryAttributes	ขึ้นอยู่กับนโยบายของผู้ให้บริการใบรับรองอิเล็กทรอนิกส์ว่าจะมีการใส่ข้อมูลใน field นี้หรือไม่
Basic Constraints	จะมีค่าเป็น False เนื่องจากไม่ใช่ใบรับรองอิเล็กทรอนิกส์ของผู้ให้บริการ
Name Constraints	ไม่มีการกำหนดค่า เนื่องจาก RFC 3280 ระบุว่าฟิลด์ nameConstraints ใช้ใน CA Certificate เท่านั้น

ตารางที่ 3.12 (ต่อ)

Policy Constraints	ไม่มีการกำหนดค่า เนื่องจาก RFC 3280 ระบุว่าฟิลด์ policyConstraints ใช้ใน CA Certificate เท่านั้น
Enhanced Key Usage	ต้องมีการระบุค่าเป็น Time Stamping เพื่อระบุว่าใบรับรองอิเล็กทรอนิกส์ใบนี้ใช้สำหรับสร้าง Time-Stamping
CRL Distribution Points	มีการระบุข้อมูลลงในฟิลด์นี้ เพื่อช่วยในการสนับสนุนการตรวจสอบสถานะของใบรับรองอิเล็กทรอนิกส์ ซึ่งขึ้นอยู่กับนโยบายของผู้ให้บริการใบรับรองอิเล็กทรอนิกส์
Inhibit Any-Policy	ไม่มีการกำหนดค่าลงในฟิลด์นี้ เนื่องจากไม่จำเป็นต้องใช้ใน End Entity Certificate

3.6.2 การออกแบบโครงสร้างใบรับรองอิเล็กทรอนิกส์สำหรับผู้ให้บริการประทับเวลา

ตารางที่ 3.13 การออกแบบโครงสร้างใบรับรองอิเล็กทรอนิกส์สำหรับผู้ให้บริการ ประทับเวลา

Field Name	ตัวอย่างข้อมูล
Version	V3
Serial Number	เลขฐานสิบหก
Signature Algorithm	sha1WithRSAEncryption
Issuer	X.500 Format
Validity	UTC Format
Subject	X.500 Format
Subject Public Key Information	Key length 1024 เป็นอย่างต่ำ
Authority Key Identifier	KeyID=41 77 e9 88 2d 9f 31 8f 95 36 dc 8d f5 90 25 b8 30 ca 63 04
Subject Key Identifier	d8 52 91 4f a1 0e 4a c1 53 ba 4d 83 a8 87 16 bb e7 66 a9 3d
Key Usage	Digital Signature

ตารางที่ 3.13 (ต่อ)

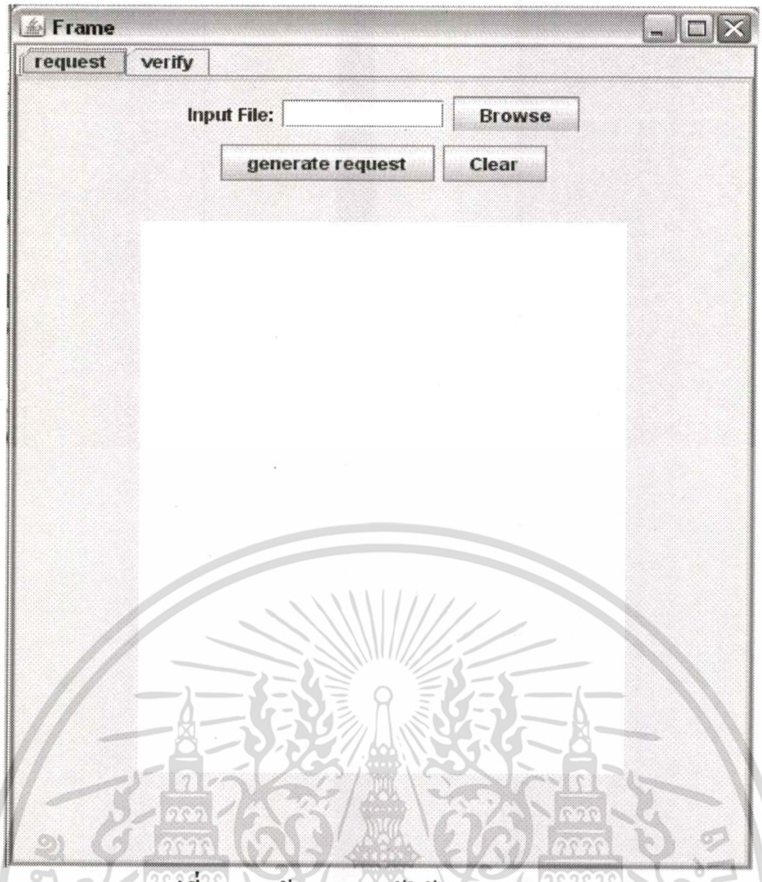
Certificate Policy	[1]Certificate Policy: Policy Identifier=1.3.6.1.4.1.19500.2.1.1 [1,1]Policy Qualifier Info: Policy Qualifier Id=CPS Qualifier: http://gca.thai gov.net
Basic Constraints	False
Enhanced Key Usage	Time stamping
CRL Distribution Points	[1]CRL Distribution Point Distribution Point Name: Full Name: URL=http://crl.gca.thaigov.net/crl/crl_gca_gpca_1.crl

3.9 การออกแบบส่วนการติดต่อกับกับผู้ใช้งาน

3.9.1 การออกแบบส่วนการติดต่อกับกับผู้ใช้งานส่วนการขอประทับเวลา

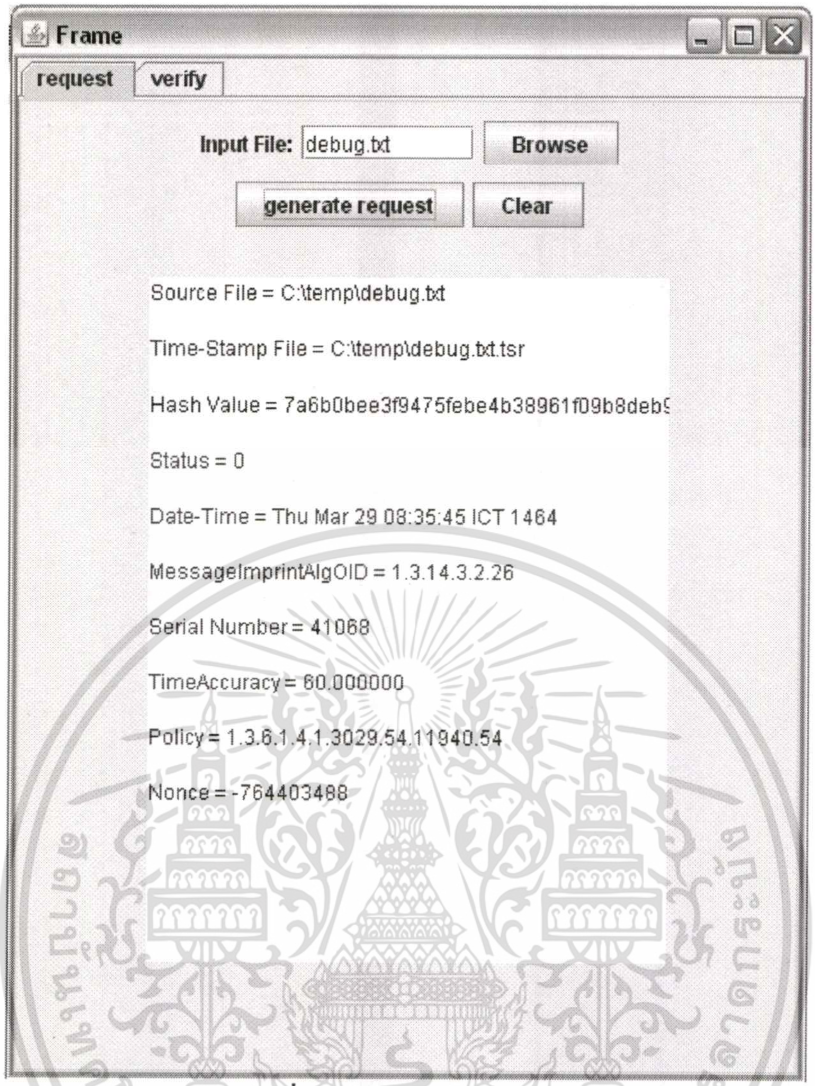
หน้าจอสำหรับการขอประทับเวลาเป็นหน้าจอสำหรับให้ผู้ใช้ทำขอประทับเวลาโดยผู้ใช้จะต้องทำการสร้างไฟล์ (Time-Stamp Request) และทำการส่งเข้าไปยังระบบ

เมื่อต้องการขอ Time-Stamp ให้เลือกที่แท็ป Request แล้วทำการ browse ข้อมูลที่ต้องการ แล้วทำการกดปุ่ม Generate Request เพื่อทำการสร้าง Time-Stamp request และส่งไปยัง TSA Server แสดงดังรูปที่ 3.8



รูปที่ 3.8 หน้าสำหรับผู้ใช้ขอประทับเวลา

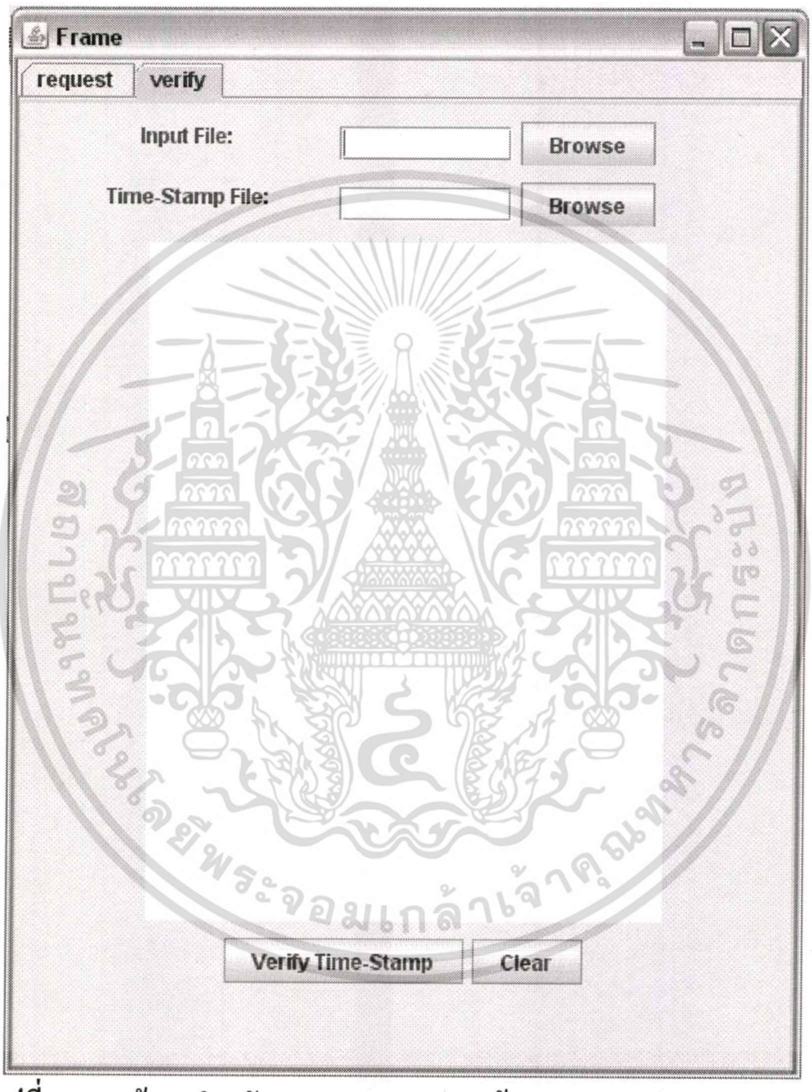
ผลการสร้าง Time-Stamp token ซึ่งจะแสดงถึงไฟล์ Time-Stamp token และรายละเอียดที่อยู่ภายใน Time-Stamp token แสดงดังรูปที่ 3.9



รูปที่ 3.9 ผลการประทับเวลา

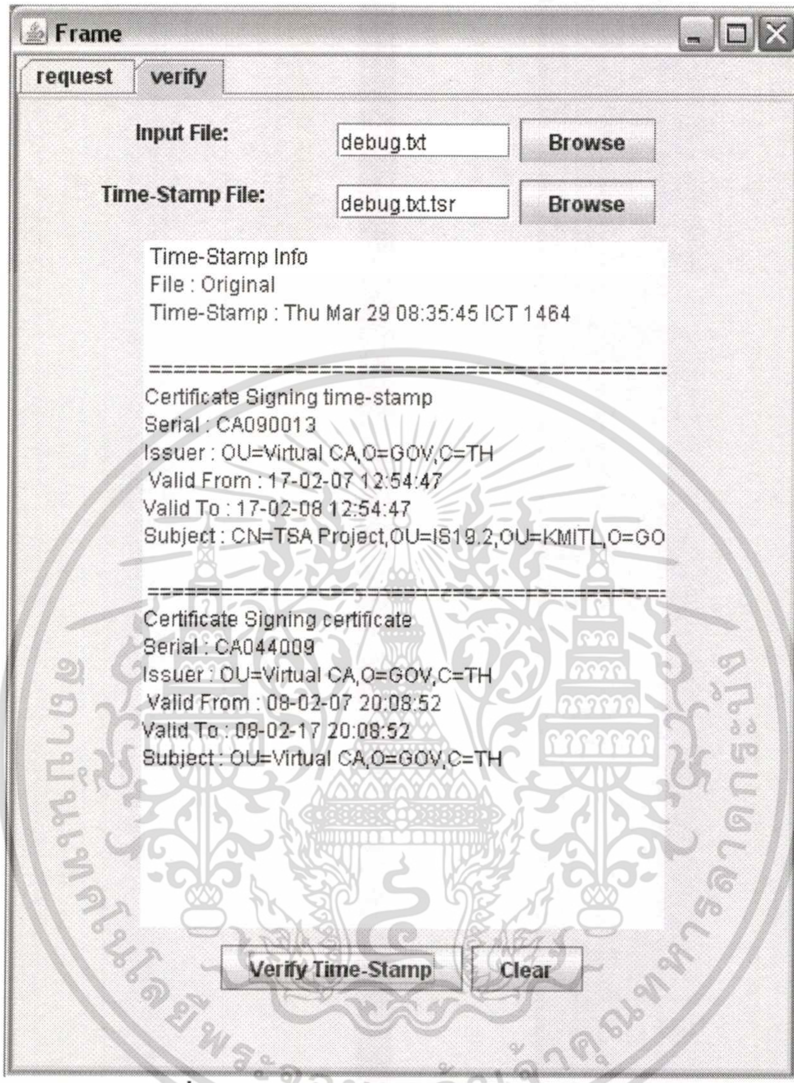
3.9.2 การออกแบบส่วนการติดต่อกับกับผู้ใช้งานส่วนการตรวจสอบข้อมูลประทับเวลา

หน้าจอสำหรับการตรวจสอบข้อมูลประทับเวลาสำหรับให้ผู้ใช้ทำตรวจสอบข้อมูลประทับเวลา เมื่อต้องการทำการตรวจสอบความถูกต้องของข้อมูลให้เลือกแท็บ Verify และทำการเลือกไฟล์ต้นฉบับและ Time-Stamp token หลังจากนั้นทำการกดปุ่ม Verify Time-Stamp เพื่อทำการตรวจสอบความถูกต้อง แสดงดังรูปที่ 3.10



รูปที่ 3.10 หน้าจอสำหรับตรวจสอบความถูกต้องของ Time-Stamp Token

ผลการตรวจสอบความถูกต้องของข้อมูลต้นฉบับเทียบกับ Time-Stamp token แสดงดังรูปที่ 3.11



รูปที่ 3.11 รายละเอียดของ Time-Stamp Token

บทที่ 4

การพัฒนาระบบ

4.1 การพัฒนา Time-Stamp Server

4.1.1 เครื่องมือที่ใช้ในการพัฒนา Time-Stamp Server

ในการพัฒนาระบบผู้ให้บริการประทับเวลานั้น จะประกอบไปด้วยการพัฒนาเซิร์ฟเวอร์ซึ่งเป็นผู้ให้บริการประทับเวลา โดยมีการใช้เครื่องมือในการพัฒนา ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 เครื่องมือที่ใช้ในการพัฒนา Time-Stamp Server

OS	Linux CentOS 4.4
Web Server	Apache 2.0.59
Web Application	PHP 5.0
Database Server	MySQL 5.0.37
Module	Openssl 0.9.8c
Patch Openssl	ts-20060923-0_9_8c
Module mod Apache	Mod-tsa-20060923

รายละเอียดของเครื่องมือที่ใช้พัฒนา

- Openssl เป็นไลบรารีสำหรับใช้ในการสร้างใบรับรองอิเล็กทรอนิกส์ และใช้ในการสร้างลายมือชื่อดิจิทัล
- Patch Openssl เป็นไลบรารีเพิ่มเติมเพื่อให้ Openssl สามารถสร้าง time-stamp token ได้
- Module mod apache เป็น module ที่ทำปรับแต่งให้ apache มีความสามารถที่จะรับ request และทำการเรียก module time-stamp เพื่อทำการสร้าง time-stamp token

4.1.2 ขั้นตอนการพัฒนา Time-Stamp Server

1. ติดตั้ง Apache ซึ่งเวอร์ชั่น Apache จะต้องสนับสนุนกับ module mod apache ด้วย
2. ทำการ patch Openssl ซึ่งเวอร์ชั่น patch Openssl จะต้องสนับสนุนกับ Openssl ด้วย
3. ติดตั้ง Openssl
4. ติดตั้ง module mod apache
5. ติดตั้ง tsa.conf ลงใน httpd.conf ของ Apache
6. ปรับแต่ง tsa.conf

4.2 การพัฒนา Module Generate Time-Stamp request

4.2.1 เครื่องมือที่ใช้ในการพัฒนา Module Generate Time-Stamp Request

ในการพัฒนาระบบผู้ให้บริการประทับเวลานั้น จะประกอบด้วยอีกส่วนหนึ่งที่สำคัญคือ Module Generate Time-Stam6p Request หรือโมดูลในการร้องขอเพื่อจะขอประทับเวลา โดยมีการใช้เครื่องมือในการพัฒนาโมดูลนี้ ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 เครื่องมือที่ใช้ในการพัฒนา Module Generate Time-Stamp Request

JDK	Java SDK 1.6
Tool	Eclipse
Library	- bouncycastle provider for jdk 1.6 - smime / cms for jdk 1.6 - tsp for jdk 1.6 - commons-httpclient library - commons-logging library - log4j library

รายละเอียดของเครื่องมือที่ใช้พัฒนา

- bouncycastle provider เป็นไลบรารี เพิ่มเติมเพื่อให้ Java มีประสิทธิภาพในด้าน Cryptographic มากยิ่งขึ้น เป็นไลบรารี พื้นฐานเพื่อใช้ library อื่น ๆ ต่อไป
- smime/cms เป็นไลบรารี เกี่ยวกับ Cryptographic Message Syntax ใช้ในการจัดการเกี่ยวกับไวยากรณ์ของ time-stamp request และ time-stamp token
- tsp เป็นไลบรารี เกี่ยวกับ Time-Stamp Protocol ใช้ในการสร้าง time-stamp request ตามมาตรฐาน RFC 3161
- commons-httpclient เป็นไลบรารีที่ใช้จัดการเกี่ยวกับฟังก์ชันของ HTTP โดยจะมี commons-logging เป็นไลบรารีที่ใช้คู่กัน
- commons-logging เป็นไลบรารีที่ใช้จัดการเกี่ยวกับฟังก์ชันการเก็บบันทึก log เกิดจากฟังก์ชันงานของ httpclient
- log4j เป็นไลบรารีที่ใช้จัดการในการบันทึก log ที่เกิดจากความผิดพลาด

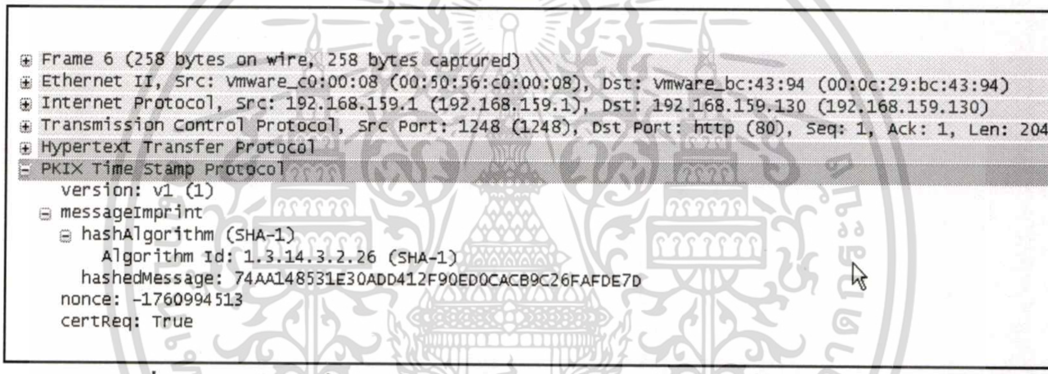
4.2.2 ขั้นตอนการพัฒนา Module generate Time-Stamp request

1. ติดตั้งไลบรารีดังกล่าวลงใน project ดังกล่าว
2. พัฒนา module generate time-stamp request
3. ทดสอบ
4. Export เป็น JAR Execute

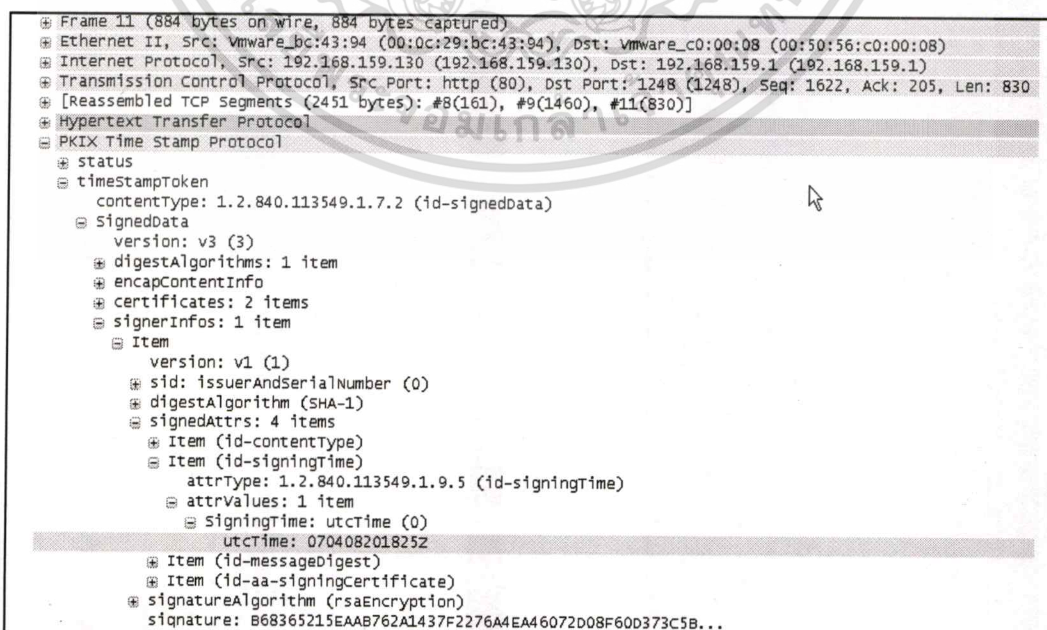
4.3 การทดสอบระบบผู้ให้บริการประทับเวลา

4.3.1 การทดสอบโดยใช้ Module Generate Time-Stamp Request

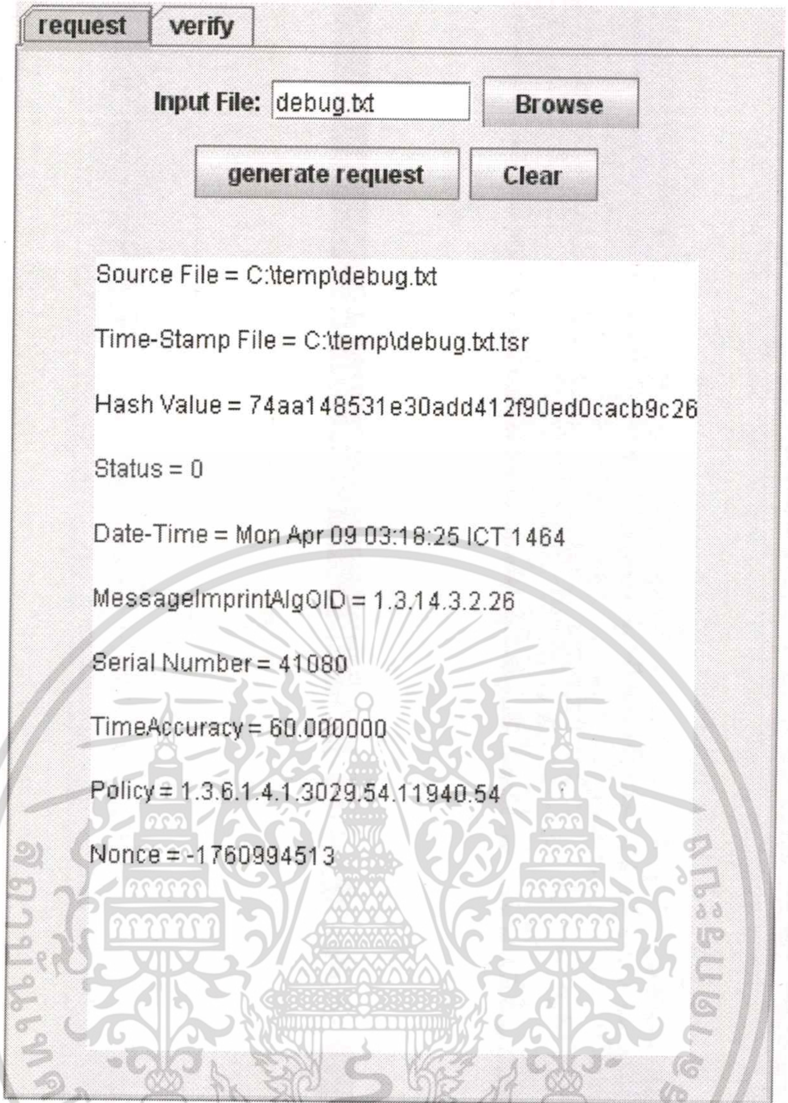
1. การทดสอบโดยส่ง request ที่ใช้อัลกอริทึมที่ใช้ในการสร้าง messageImprint ที่ระบบได้อนุญาติ ไม่ได้ทำการร้องขอ Policy ID ใดๆ ซึ่งจะแสดงผลดังรูปที่ 4.1 - 4.3



รูปที่ 4.1 Request ที่มี hashAlgorithm SHA-1 และไม่มีการร้องขอ Policy ID



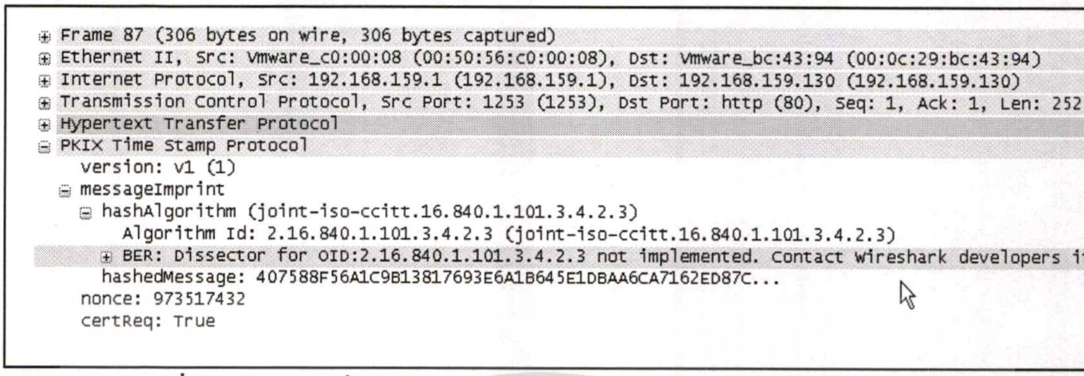
รูปที่ 4.2 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 1



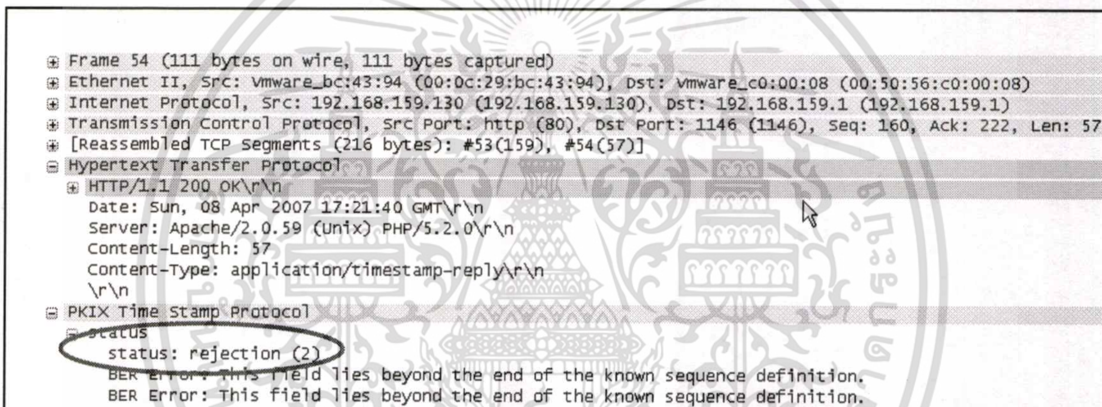
รูปที่ 4.3 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 1 ใน user interface

โดยในการทดสอบนี้ได้ทำการส่ง Time-Stamp request ที่ใช้ SHA-1 ในการสร้าง messageImprint และไม่มีการร้องขอ Policy ID ซึ่งเมื่อ Server ทำการตรวจสอบเสร็จแล้ว จึงทำการสร้าง Time-Stamp token มีค่าสถานะเท่ากับ 1 มีค่า nonce ใน time-stamp token เท่ากับค่า nonce ที่อยู่ใน time-stamp token ซึ่งถูกต้องตามกระบวนการที่ได้ออกแบบไว้

2. การทดสอบโดยส่ง request ที่ใช้อัลกอริทึมที่ใช้ในการสร้าง messageImprint ที่ระบบไม่ได้อนุญาติ, ไม่ได้ทำการร้องขอ Policy ID ใด ๆ ซึ่งจะแสดงผลดังรูปที่ 4.4 - 4.7



รูปที่ 4.4 request ที่มี hashAlgorithm SHA-512 และไม่มีการร้องขอ Policy ID



รูปที่ 4.5 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 2

รูปที่ 4.6 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 2 ใน user interface

```
[Mon Apr 09 00:16:58 2007] [error] mod_tsa:3657:error:2D08507D:time stamp routin
es:TS_RESP_GET_POLICY:unacceptable policy:ts_resp_sign.c:591:
[Mon Apr 09 00:16:58 2007] [error] mod_tsa:3657:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
[Mon Apr 09 00:20:23 2007] [error] mod_tsa:3659:error:2D08507D:time stamp routin
es:TS_RESP_GET_POLICY:unacceptable policy:ts_resp_sign.c:591:
[Mon Apr 09 00:20:23 2007] [error] mod_tsa:3659:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
[Mon Apr 09 00:21:40 2007] [error] mod_tsa:3662:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
```

รูปที่ 4.7 Log apache เมื่อระบบได้รับ request ที่ได้ในการทดสอบข้อที่ 2

จากผลการทดสอบ ระบบไม่สามารถสร้าง time-stamp token ได้เนื่องจากในการทดสอบข้อที่ 2 นี้ ได้เลือกใช้อัลกอริทึม SHA-512 ในการสร้าง messageImprint ซึ่งระบบอนุญาติเพียง SHA-1, MD5 ซึ่งสถานะของผลการตอบกลับจะเท่ากับ 2 ดังรูปที่ 4.5

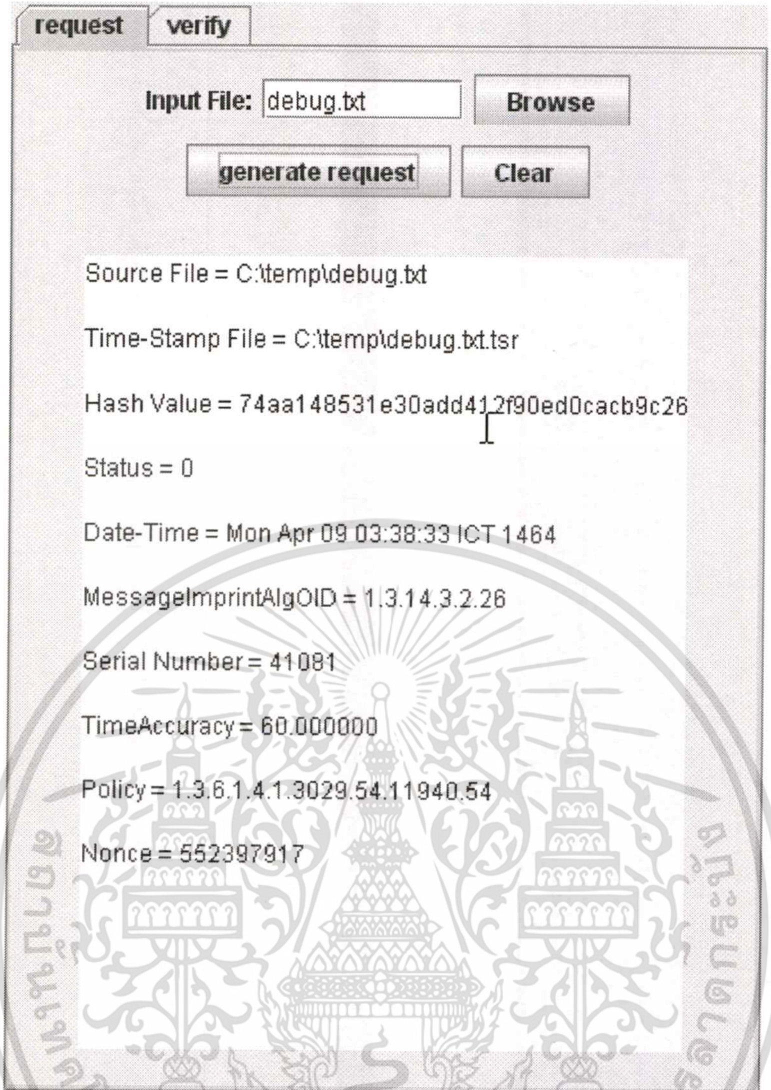
3. การทดสอบโดยส่ง request ที่ใช้อัลกอริทึมที่ใช้ในการสร้าง messageImprint ที่ระบบได้อนุญาติ, ทำการร้องขอ Policy ID ที่ระบบได้อนุญาติ ซึ่งจะแสดงผลดังรูปที่ 4.8 - 4.10

```
⊞ Frame 105 (271 bytes on wire, 271 bytes captured)
⊞ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_bc:43:94 (00:0c:29:bc:43:94)
⊞ Internet Protocol, Src: 192.168.159.1 (192.168.159.1), Dst: 192.168.159.130 (192.168.159.130)
⊞ Transmission Control Protocol, Src Port: 1255 (1255), Dst Port: http (80), Seq: 1, Ack: 1, Len: 217
⊞ Hypertext Transfer Protocol
⊟ PKIX Time Stamp Protocol
    version: v1 (1)
  ⊟ messageImprint
    ⊟ hashAlgorithm (SHA-1)
        Algorithm Id: 1.3.14.3.2.26 (SHA-1)
      hashedMessage: 74AA148531E30ADD412F90ED0CACB9C26FAFDE7D
    reqPolicy: 1.3.6.1.4.1.3029.54.11940.54 (SNMPv2-SMI::enterprises.3029.54.11940.54)
    nonce: 552397917
    certReq: True
```

รูปที่ 4.8 request ที่มี hashAlgorithm SHA-1 และการร้องขอ Policy ID ที่ระบบอนุญาต

```
⊞ Frame 110 (884 bytes on wire, 884 bytes captured)
⊞ Ethernet II, Src: Vmware_bc:43:94 (00:0c:29:bc:43:94), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
⊞ Internet Protocol, Src: 192.168.159.130 (192.168.159.130), Dst: 192.168.159.1 (192.168.159.1)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 1255 (1255), Seq: 1622, Ack: 218, Len: 830
⊞ [Reassembled TCP Segments (2451 bytes): #107(161), #108(1460), #110(830)]
⊞ Hypertext Transfer Protocol
⊟ PKIX Time Stamp Protocol
  ⊞ status
  ⊟ timeStampToken
      contentType: 1.2.840.113549.1.7.2 (id-signedData)
    ⊟ SignedData
        version: v3 (3)
      ⊞ digestAlgorithms: 1 item
      ⊞ encapContentInfo
      ⊞ certificates: 2 items
      ⊟ signerInfos: 1 item
        ⊟ Item
            version: v1 (1)
          ⊞ sid: issuerAndSerialNumber (0)
          ⊞ digestAlgorithm (SHA-1)
          ⊟ signedAttrs: 4 items
            ⊞ Item (id-contentType)
            ⊟ Item (id-signingTime)
                attrType: 1.2.840.113549.1.9.5 (id-signingTime)
              ⊟ attrValues: 1 item
                ⊟ SigningTime: utcTime (0)
                    utcTime: 070408203833Z
            ⊞ Item (id-messageDigest)
            ⊞ Item (id-aa-signingCertificate)
          ⊞ signatureAlgorithm (rsaEncryption)
          signature: 4571E0E702C81FBF0A5E7AA0A96FE3EE3D997C291B8C55EB...
```

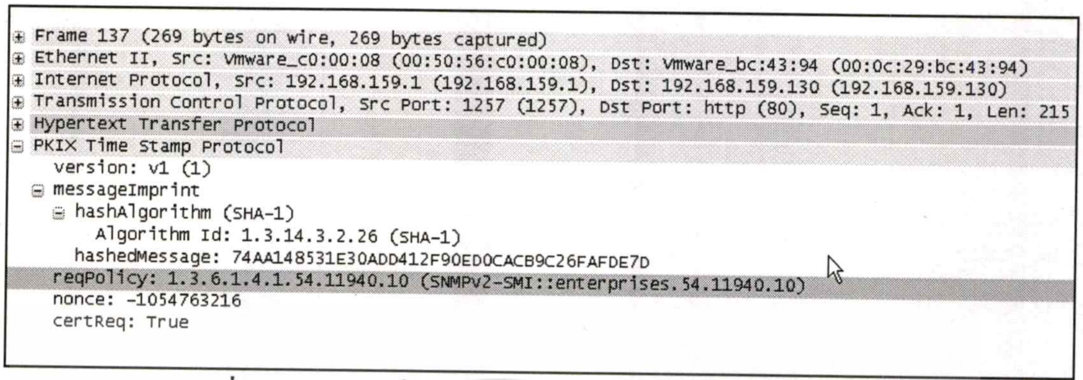
รูปที่ 4.9 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 3



รูปที่ 4.10 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 3 ใน user interface

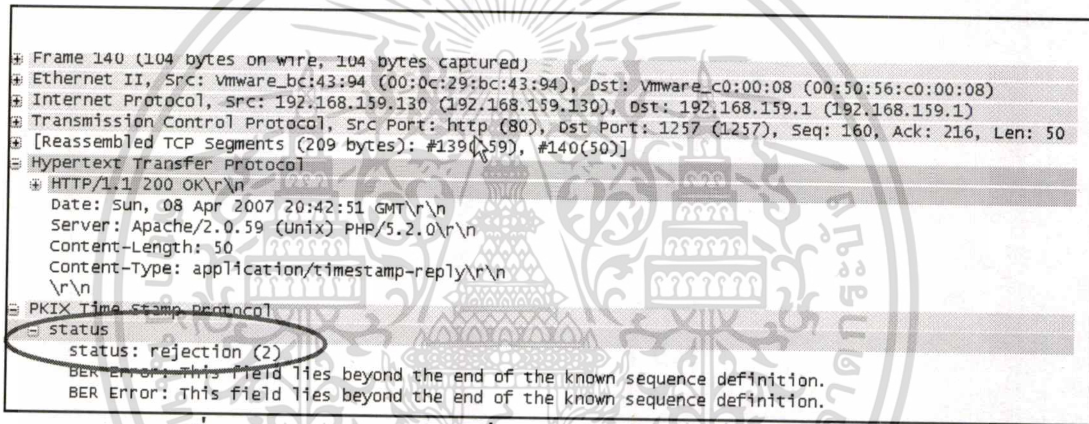
ในการทดลองข้อ 3 นี้จะใช้ SHA-1 เป็นอัลกอริทึมในการสร้าง messageImprint และทำการร้องขอ Policy ID “1.3.6.1.3029.54.11940.54” ดังรูปที่ 4.8 ซึ่งเป็น Policy ID ที่ระบบอนุญาติไว้ เมื่อระบบทำการตรวจสอบ Time-Stamp request แล้วจึงได้ทำการสร้าง Time-Stamp Token และทำการส่งกลับมายังผู้ร้องขอ แสดงดังรูปที่ 4.9 และ รูปที่ 4.10

4. การทดสอบโดยส่ง request ที่ใช้อัลกอริทึมที่ใช้ในการสร้าง messageImprint ที่ระบบได้อนุญาติ ทำการร้องขอ Policy ID ที่ระบบไม่ได้อนุญาติ ซึ่งจะแสดงผลดังรูปที่ 4.11 - 4.14



```
⊞ Frame 137 (269 bytes on wire, 269 bytes captured)
⊞ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_bc:43:94 (00:0c:29:bc:43:94)
⊞ Internet Protocol, Src: 192.168.159.1 (192.168.159.1), Dst: 192.168.159.130 (192.168.159.130)
⊞ Transmission Control Protocol, Src Port: 1257 (1257), Dst Port: http (80), Seq: 1, Ack: 1, Len: 215
⊞ Hypertext Transfer Protocol
⊟ PKIX Time Stamp Protocol
    version: v1 (1)
  ⊟ messageImprint
    ⊟ hashAlgorithm (SHA-1)
        Algorithm Id: 1.3.14.3.2.26 (SHA-1)
      hashedMessage: 74AA148531E30ADD412F90ED0CACB9C26FAFDE7D
    reqPolicy: 1.3.6.1.4.1.54.11940.10 (SNMPv2-SMI::enterprises.54.11940.10)
    nonce: -1054763216
    certReq: True
```

รูปที่ 4.11 request ที่มีการร้องขอ Policy ID ที่ระบบไม่อนุญาต



```
⊞ Frame 140 (104 bytes on wire, 104 bytes captured)
⊞ Ethernet II, Src: Vmware_bc:43:94 (00:0c:29:bc:43:94), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
⊞ Internet Protocol, Src: 192.168.159.130 (192.168.159.130), Dst: 192.168.159.1 (192.168.159.1)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 1257 (1257), Seq: 160, Ack: 216, Len: 50
⊞ [Reassembled TCP Segments (209 bytes): #139(159), #140(50)]
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    Date: Sun, 08 Apr 2007 20:42:51 GMT\r\n
    Server: Apache/2.0.59 (Unix) PHP/5.2.0\r\n
    Content-Length: 50
    Content-Type: application/timestamp-reply\r\n
    \r\n
⊟ PKIX Time Stamp Protocol
  ⊟ status
      status: rejection (2)
      BER Error: This field lies beyond the end of the known sequence definition.
      BER Error: This field lies beyond the end of the known sequence definition.
```

รูปที่ 4.12 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 4

รูปที่ 4.13 time-stamp token ที่ได้จาก request ในการทดสอบข้อที่ 3 ใน user interface

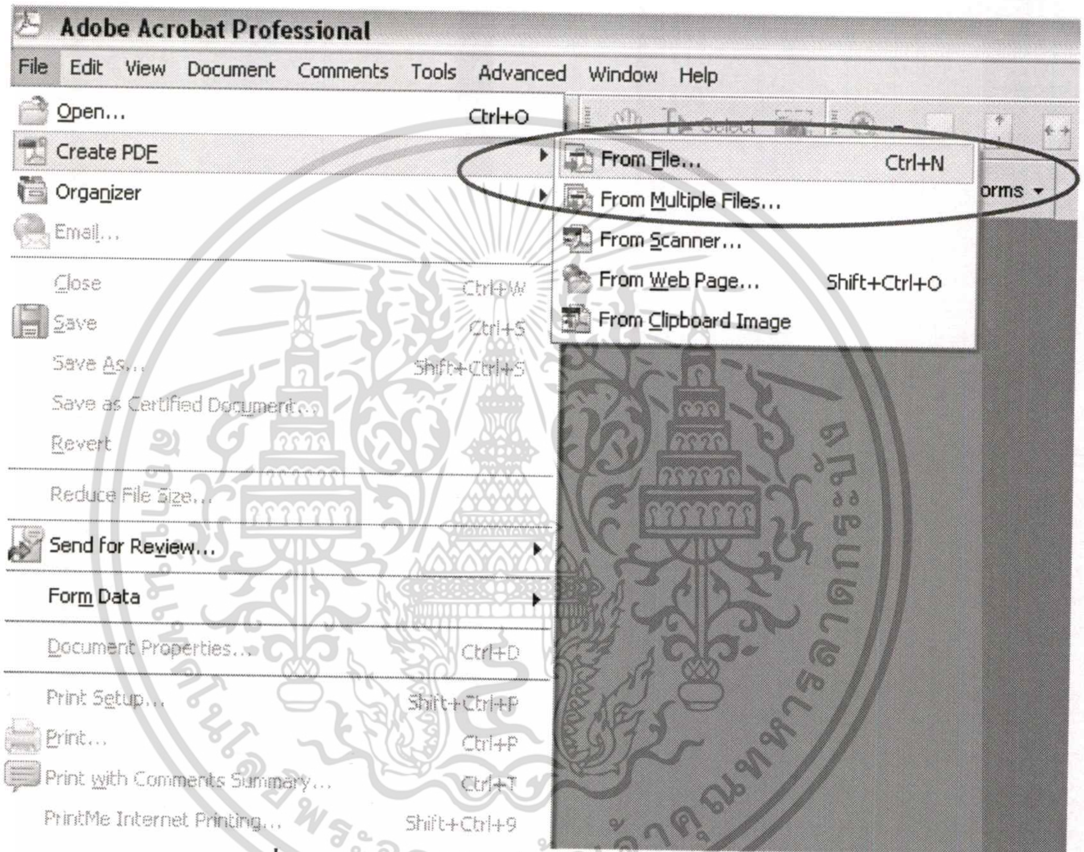
```
[Mon Apr 09 00:16:58 2007] [error] mod_tsa:3657:error:2D08507D:time stamp routin
es:TS_RESP_GET_POLICY:unacceptable policy:ts_resp_sign.c:591:
[Mon Apr 09 00:16:58 2007] [error] mod_tsa:3657:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
[Mon Apr 09 00:20:23 2007] [error] mod_tsa:3659:error:2D08507D:time stamp routin
es:TS_RESP_GET_POLICY:unacceptable policy:ts_resp_sign.c:591:
[Mon Apr 09 00:20:23 2007] [error] mod_tsa:3659:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
[Mon Apr 09 00:21:40 2007] [error] mod_tsa:3662:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
[Mon Apr 09 03:36:32 2007] [error] mod_tsa:3657:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
[Mon Apr 09 03:42:51 2007] [error] mod_tsa:3659:error:2D08507D:time stamp routin
es:TS_RESP_GET_POLICY:unacceptable policy:ts_resp_sign.c:591:
[Mon Apr 09 03:42:51 2007] [error] mod_tsa:3659:error:2D07A079:time stamp routin
es:TS_RESP_create_response:response setup error:ts_resp_sign.c:468:
```

รูปที่ 4.14 log apache เมื่อระบบได้รับ request ที่ได้ในการทดสอบข้อที่ 4

จากผลการทดสอบ ระบบไม่สามารถสร้าง time-stamp token ได้เนื่องจากในการทดสอบข้อที่ 4 นี้ ได้ทำการร้องขอ Policy ID “1.3.6.1.4.1.54.11940.10” ซึ่งระบบอนุญาติให้ขอ Policy ID “1.3.6.1.4.1.54.11940.54” ซึ่งสถานะของผลตอบกลับเท่ากับ 2 ดังรูปที่ 4.12

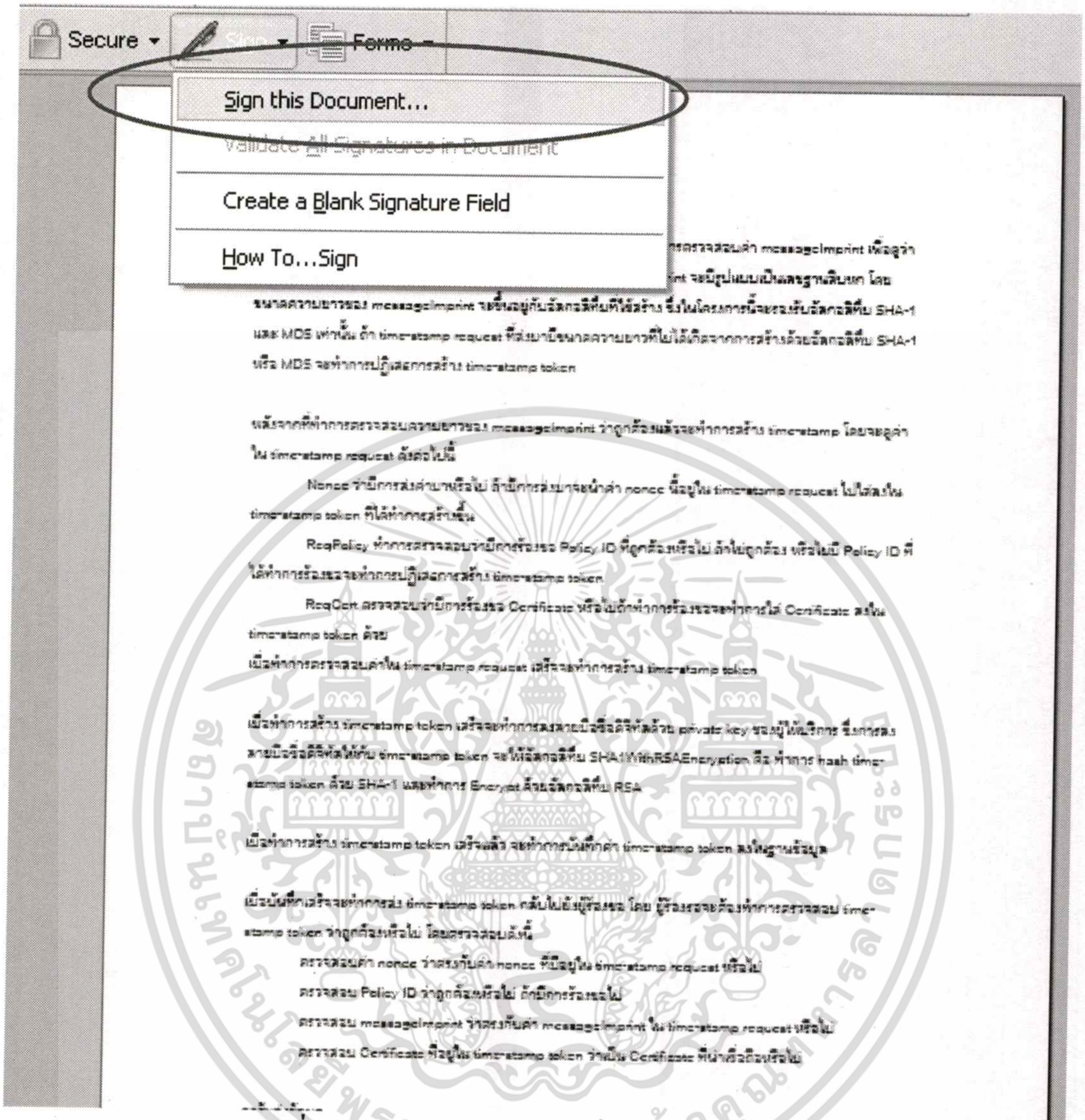
4.3.1 การทดสอบโดยใช้ Adobe Acrobat Professional 7.0

1. ทำการสร้างไฟล์ PDF



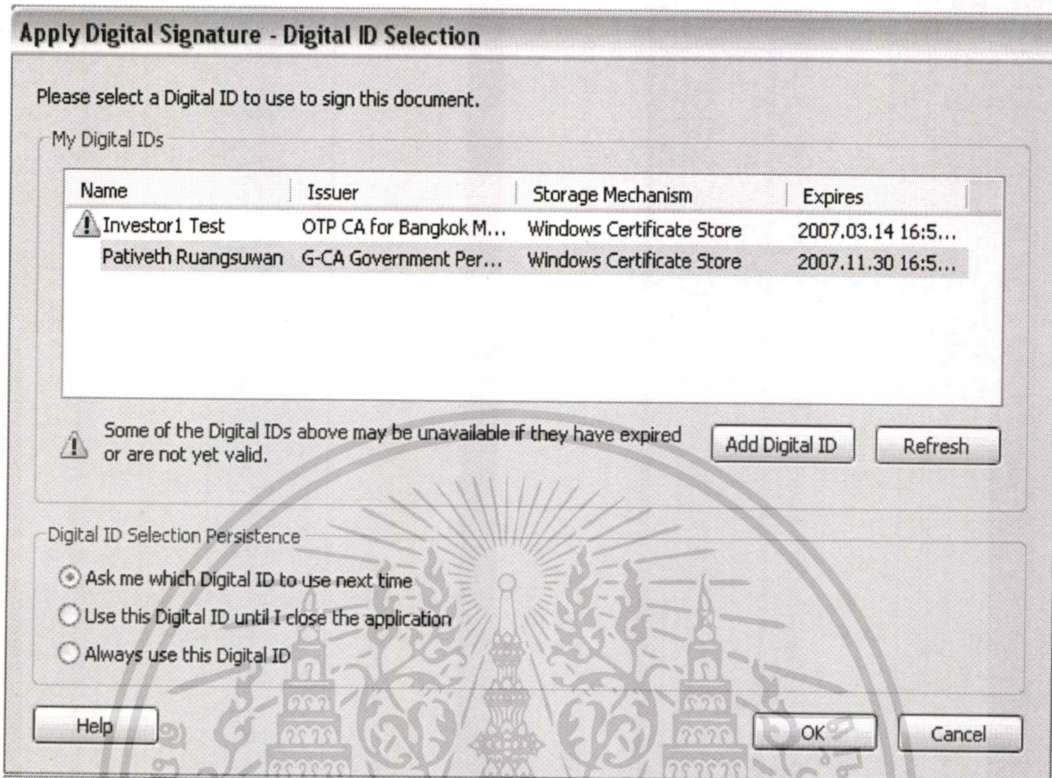
รูปที่ 4.15 การเลือกไฟล์เพื่อนำไปสร้างเป็นไฟล์ PDF

2. เมื่อสร้างไฟล์ PDF แล้วทำการลงลายมือชื่อดิจิทัลให้กับไฟล์ PDF



รูปที่ 4.16 แสดงการลงลายมือชื่อดิจิทัลให้กับไฟล์ PDF

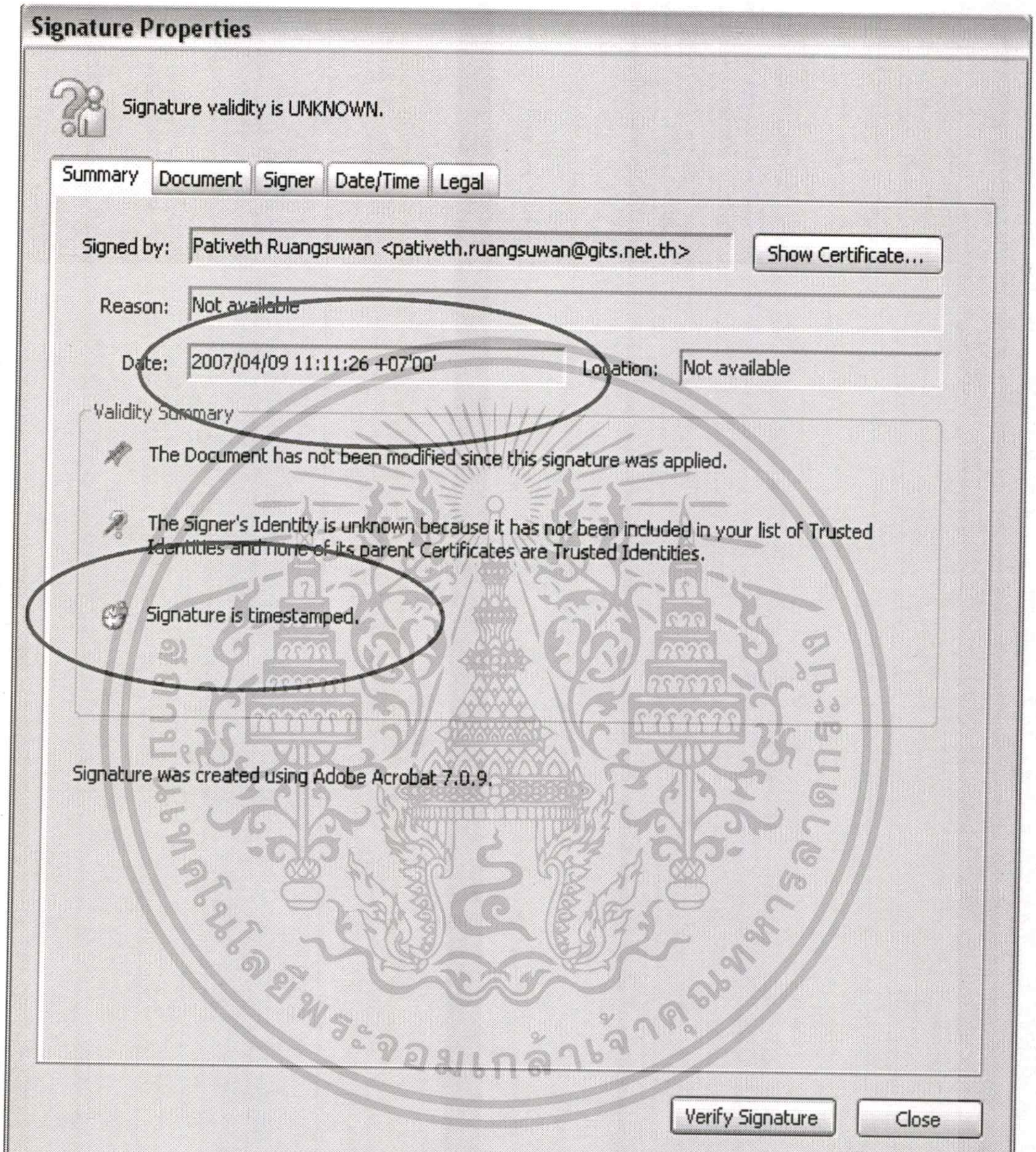
3. ทำการเลือกใบรับรองอิเล็กทรอนิกส์เพื่อลงลายมือชื่อดิจิทัล



รูปที่ 4.17 การเลือกใบรับรองอิเล็กทรอนิกส์เพื่อทำการลงลายมือชื่อดิจิทัล

4. แสดงผลการลงลายมือชื่อดิจิทัลพร้อมกับลง Time-Stamp

รายละเอียดของรายมือชื่อดิจิทัลที่มีการประทับเวลาแสดงดังรูปที่ 4.17



รูปที่ 4.18 รายละเอียดของ Time-Stamp ของลายมือชื่อดิจิทัล

ข้อมูลใน Time-Stamp request ที่เกิดจากการร้องขอของ Acrobat ซึ่งใช้ SHA-1 เป็นอัลกอริทึมในการสร้าง messageImprint แสดงดังรูป 4.19

```
⊞ Frame 6 (327 bytes on wire, 327 bytes captured)
⊞ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_bc:43:94 (00:0c:29:bc:43:94)
⊞ Internet Protocol, Src: 192.168.159.1 (192.168.159.1), Dst: 192.168.159.130 (192.168.159.130)
⊞ Transmission Control Protocol, Src Port: 1853 (1853), Dst Port: http (80), Seq: 1, Ack: 1, Len: 273
⊟ Hypertext Transfer Protocol
  ⊞ POST /tsa/ HTTP/1.1\r\n
    Accept: */*\r\n
    Content-Type: application/timestamp-query\r\n
    Content-Length: 53
    Character-Encoding: binary\r\n
    User-Agent: PPKHandler\r\n
    Host: tsa.proj.dev\r\n
    Connection: Keep-Alive\r\n
    Cache-Control: no-cache\r\n
    \r\n
⊟ PKIX Time Stamp Protocol
    version: v1 (1)
  ⊟ messageImprint
    ⊟ hashAlgorithm (SHA-1)
        Algorithm Id: 1.3.14.3.2.26 (SHA-1)
      hashedMessage: F45B74FAD2858A9FC2E912BA944F89CB55FF6B1A
    nonce: 503460339
    certReq: True
```

รูปที่ 4.19 Time-Stamp request ที่เกิดจาก Acrobat

แสดงข้อมูลใน Time-Stamp token ที่สร้างจาก Time-Stamp request ของ Acrobat ซึ่งมีไวยากรณ์ของ Time-Stamp request ที่ถูกต้อง ดังรูปที่ 4.20

```
⊞ Frame 11 (888 bytes on wire, 888 bytes captured)
⊞ Ethernet II, Src: Vmware_bc:43:94 (00:0c:29:bc:43:94), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
⊞ Internet Protocol, Src: 192.168.159.130 (192.168.159.130), Dst: 192.168.159.1 (192.168.159.1)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 1853 (1853), Seq: 1679, Ack: 274, Len: 834
⊞ [Reassembled TCP Segments (2512 bytes): #8(218), #9(1460), #11(834)]
⊞ Hypertext Transfer Protocol
⊟ PKIX Time Stamp Protocol
  ⊟ status
      status: granted (0)
  ⊟ timeStampToken
      contentType: 1.2.840.113549.1.7.2 (id-signedData)
    ⊟ SignedData
        version: v3 (3)
      ⊞ digestAlgorithms: 1 item
      ⊞ encapContentInfo
      ⊞ certificates: 2 items
      ⊟ signerInfos: 1 item
        ⊟ Item
            version: v1 (1)
          ⊞ sid: issuerAndSerialNumber (0)
          ⊞ digestAlgorithm (SHA-1)
          ⊟ signedAttrs: 4 items
            ⊞ Item (id-contentType)
            ⊟ Item (id-signingTime)
                attrType: 1.2.840.113549.1.9.5 (id-signingTime)
              ⊟ attrValues: 1 item
                ⊞ SigningTime: utcTime (0)
            ⊞ Item (id-messageDigest)
            ⊞ Item (id-aa-signingCertificate)
          ⊞ signatureAlgorithm (rsaEncryption)
            signature: 9DF593EAE7C748891684742C1203D242F44CA3E69FCF961B...
```

รูปที่ 4.20 Time-Stamp token ที่เกิดจากร้องขอจาก Acrobat

4.3.3 สรุปผลการทดสอบ

จากการทดสอบสรุปได้ว่า Time-Stamp Sever สามารถให้บริการประทับเวลาได้ก็ต่อเมื่อ Time-Stamp request นั้นถูกต้องตามที่ผู้ให้บริการกำหนดไว้ และไม่สามารถให้บริการประทับเวา ได้ถ้า Time-Stamp request นั้นไม่ถูกต้องตามที่ผู้ให้บริการกำหนดไว้ ซึ่งตรงตามมาตรฐาน RFC 3161 (Internet X.509 Public Key Infrastructure: Time-Stamp Protocol)

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปโครงการ

ในโครงการนี้ได้ดำเนินการศึกษาข้อมูล ทฤษฎี และมาตรฐาน RFC 3280 (Internet X.509 Public Key Infrastructure: Certificate and CRL Profile) เพื่อทำการสร้างใบรับรองอิเล็กทรอนิกส์สำหรับ Time-Stamping ,มาตรฐาน RFC 3161 (Internet X.509 Public Key Infrastructure: Time-Stamp Protocol) เพื่อทำการออกแบบระบบผู้ให้บริการประทับเวลา โดยในระหว่างขั้นตอนการวิเคราะห์และออกแบบโปรแกรมก็ได้ทำการศึกษาเกี่ยวกับ UML เพื่อออกแบบระบบให้ถูกต้อง และได้นำทฤษฎีแนวคิดเทคโนโยลีต่าง ๆ ที่เกี่ยวข้องมาพัฒนาเป็นระบบ เพื่อให้ได้ระบบสำหรับให้บริการประทับเวลา เพื่อนำไปพัฒนาและสามารถนำไปประยุกต์เข้ากับแอปพริเคชั่น หรือ ระบบให้บริการต่าง ๆ ได้เช่นระบบจ่ายเงิน ระบบใบลา ได้ หรือระบบที่ต้องการความถูกต้องของเวลา

5.2 ข้อจำกัดของระบบที่พัฒนา

1. ระบบนี้พัฒนาเซิฟเวอร์สำหรับให้บริการประทับเวลาแต่เนื่องด้วยจากแอปพริเคชั่นที่สามารถรองรับเพื่อให้เห็นการใช้งานจริงนั้นยังไม่มีมาก ในการพัฒนาจึงได้ใช้โปรแกรม Adobe Acrobat เพื่อใช้ในการทดสอบเพื่อให้เห็นการใช้งานอย่างแท้จริง

2. ระบบให้บริการประทับเวลาใช้แอปพริเคชั่นแบบโอเพนซอร์สในการพัฒนา จึงอาจจะมีการเกิดความผิดพลาดได้บ้างเนื่องจากแอปพริเคชั่นที่ใช้ไม่มีการรองรับมาตรฐาน

3. การพัฒนาอยู่บนสมมุติฐานว่าระบบอยู่ในพื้นที่ปลอดภัย และมีเฉพาะผู้ที่มีสิทธิเท่านั้นที่สามารถเข้าถึงระบบได้

5.3 ปัญหาและอุปสรรคระหว่างพัฒนาระบบ

1. การติดตั้งระบบให้บริการประทับเวลาค่อนข้างยาก เนื่องจากเป็นแอปพริเคชั่น เฉพาะทาง ไม่มีคู่มือในการใช้งาน ต้องอาศัยการทดลองหลาย ๆ ครั้งจึงประสบผลสำเร็จ

2. ไม่มีแอปพริเคชั่นที่รองรับเท่าที่ควรจึงทำให้การเข้าใจการทำงานของแอปพริเคชั่นค่อนข้างลำบาก

3. ทำการ Synchronize Time กับ NTP Server เท่านั้นเนื่องจากเครื่องที่สามารถให้บริการได้เที่ยงตรง หรือการทำSynchronize กับดาวเทียมนั้นค่อนข้างมีค่าใช้จ่ายสูง

5.4 ข้อเสนอแนะในการพัฒนาต่อ

1. พัฒนาในส่วนของการรับส่งข้อมูลประเภทอีเมล์แอดเดรส , ผ่าน Socket , ผ่าน FTP
2. พัฒนา Module กลางเพื่อนำไปประยุกต์เข้ากับระบบให้บริการต่าง ๆ
3. พัฒนาใส่ส่วนความปลอดภัย เพื่อให้ระบบมีความน่าเชื่อถือ

บรรณานุกรม

ชาลี วรกุลพิพัฒน์ และเทพฤทธิ์ บัณฑิตวัฒนาวงศ์. 2544. **UML ภาษามาตรฐานเพื่อผู้พัฒนาซอฟแวร์.** Se-Education Public Company Limited.

ปฏิเวธ เรืองสุวรรณ. 2549 "ระบบผู้ให้บริการประทับเวลา." สัมมนา 2. 2549

C.Adams, P.Cain and D.Pinkas. 2001. **"Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)."** [online] available : http://www.ietf.org/rfc/rfc3161.txt

R. Housley. 2002. **"Cryptographic Message Syntax (CMS), RFC 3369."** [online] available : http://www.ietf.org/rfc/rfc3369.txt

R. Housley. 2002. **"Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile"** [online] available : http://www.ietf.org/rfc/rfc3280.txt

Stuart Haber and W.Scott Stornetta. 1991. **"How to Time-Stamp a Digital Document."** [online] available : http://citeseer.ist.psu.edu/cache/papers/cs/11148/ http:zSzzSzwww.star-lab.comzSzhaberzSzHS-how-to-TS.pdf/haber91how.pdf

Youyong Zou n.d. 2006. **"Implement TSP Protocol."** [online] available : http://www.cs.umbc.edu/~yzou1/681report.pdf

ภาคผนวก ก
ขั้นตอนการติดตั้งโปรแกรม

1. การติดตั้งโปรแกรมระบบ Time-Stamp Server

1.1 ติดตั้ง Web Server Apache

ทำการแตกไฟล์ tar xvfz httpd-2.0.59.tar.gz ตามคำสั่งข้างล่าง

ขั้นตอนการ config เพื่อที่จะทำการ install

```
. ./configure --prefix=/usr/local/apache \
    --enable-so \
    --enable-cgi \
    --enable-info \
    --enable-rewrite \
    --enable-speling \
    --enable-usertrack \
    --enable-deflate \
    --enable-ssl \
    --enable-mime-magic
```

ใช้คำสั่ง install , make และ make install ลำดับ

1.2 ทำการ patch openssl และ install openssl

ทำการแตกไฟล์ openssl และ patch ts ที่ source ของ openssl

```
gzip -cd openssl-0.9.8c.tar.gz | tar xf –
gzip -cd ts-20060923-0_9_8c-patch.gz | patch -p1
```

configure เพื่อทำการติดตั้ง openssl

```
./config –prefix=/usr/local/openssl
```

ใช้คำสั่ง install , make และ make install ลำดับ

1.3 ทำการ mod Apache

ทำการแตกไฟล์ mod_tsa

gzip -cd mod_tsa-20060923.tar | tar xf –

ทำการ install make OPENSSL=/usr/local/openssl

1.4 ทำการ configure Apache

แทรก include /home/TSA/conf/tsa.conf ในไฟล์ httpd.conf ของ Apache

1.5 ทำการ configure tsa.conf ซึ่งสามารถทำได้ตามตัวอย่าง

```
<IfModule mod_tsa.c>

#
# Assign the tsa handler to a URL.
#
<Location /tsa>
   SetHandler tsa
   Order allow,deny
   Allow from all
</Location>

#
# File for storing serial numbers for time stamp response generation.
# The file is updated every time a response is generated.
# If the file does not exist mod_tsa tries to create and initialise
# it with 0. Default is conf/tsaserial.
#
 TSASerialFile /home/TSA/pointer/tsaserial
#
# The identifier of the OpenSSL engine to use for cryptographic
operations.
# The signing private key must be available on this engine. Any engine
types
# supported by OpenSSL can be specified here, specify "builtin" for
# the default software engine. Default is builtin.
#
```

```
# Certificate path for the TSA signing certificate excluding the TSA cert.
# The TSA certificate and these certificates are included in the SignedData
# if the time stamp query requests it. Default is the empty certificate path.
#
TSACertificateChain /home/TSA/chain/ca.pem
#
# The private key of the TSA, it must match the public key of
# the TSA certificate. Default is conf/tsakey.pem.
#
TSAKey /home/TSA/key/TSA_KEY_DECRYPT.pem
#
# This switch specifies if the TSA private key is protected by a passphrase.
# If this switch is turned on a passphrase must be entered when mod_tsa is
# being initialised. Default is Off.
#
TSAKeyPassPhrase OFF
#
# The OID of the default TSA policy that is used in responses
# if the request does not specify any policy. There is no default.
#
TSADefaultPolicy 1.3.6.1.4.1.3029.54.11940.54
#
# List of TSA policy OIDs that are acceptable by the TSA in addition to
# the default policy. These are returned only if the client explicitly
# requests one of them. The default is empty list.
#
TSAPolicies 1.1.3 1.1.4
# List of message digests that the TSA accepts. The supported algorithms:
# md2 md4 md5 sha sha1 mdc2 ripemd160. Default is sha1.
#
TSAMessageDigests sha1 md5
#
```

```
# Accuracy of the TSA time source in seconds, milliseconds and microseconds.
# Default is 0 0 0.
#
TSAAccuracy 60 0 0
#
# Specifies the maximum number of digits, which represent the fraction of
# seconds, that  need to be included in the time field. The trailing zeroes
# must be removed from the time, so there might actually be fewer digits,
# or no fraction of seconds at all. The maximum permitted value is 6.
# Default is 0.
#
TSAClockPrecisionDigits 0
#
# Specifies whether the ordering field should be set in time stamp responses.
# Default is Off.
#
TSAOrdering Off
#
# Specifies if the TSA name should be included in the responses.
# If this switched is turned on the subject name of the TSA certificate will be
# included in the time stamp response as the TSA name. Default is Off.
#
TSAIncludeName On
#
# Specifies if ESS certificate identifiers should be included for
# the certificate chain as well beside the signing certificate. The certificate
# identifier list is generated based on the certificates specified by
# TSACertificateChain. If this flag is turned off only the signing certificate
# identifier is included in the ESS SigningCertificate signed attribute.
# Default is Off.
TSAESSCertIdChain On
```

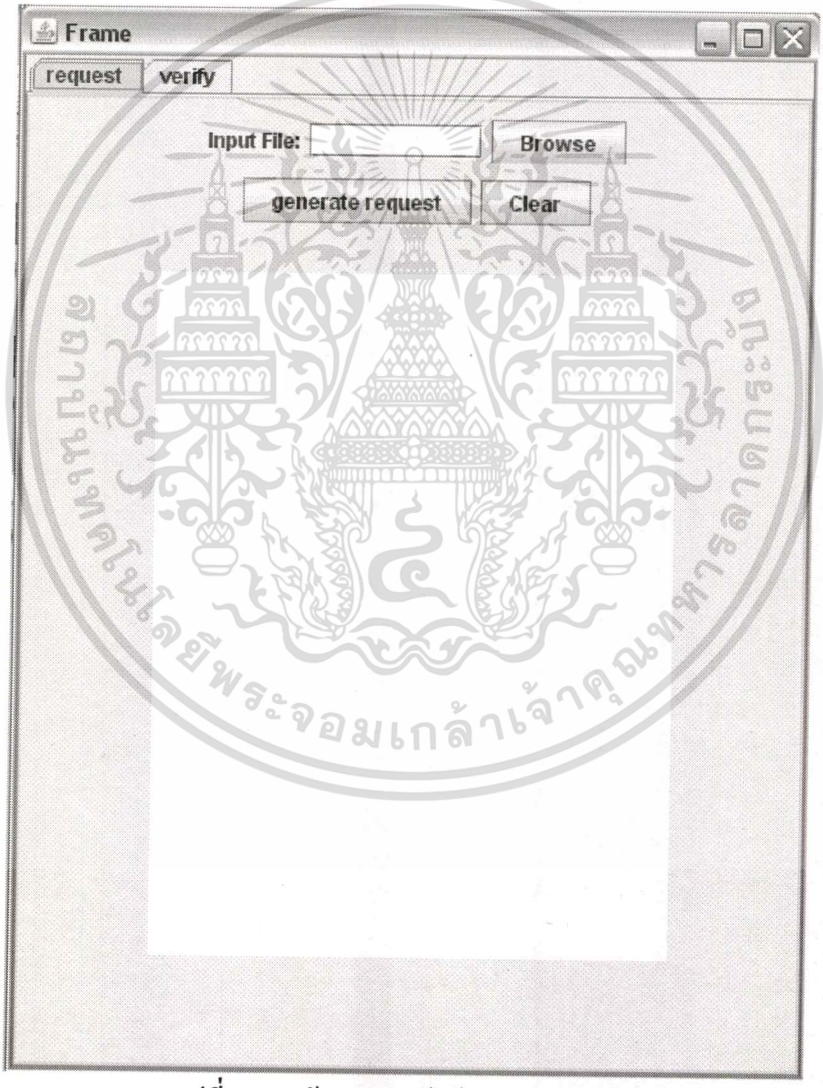
```
# Specifies the module used for storing the time stamp tokens. Available
# options: None, MySQL, FireBird, PostgreSQL. Default is None.
#
 TSADBModule MySQL
#
# (Only if TSADBModule is MySQL) Specifies the IP address or DNS of the MySQL
# host used for storing the time stamp tokens. Default is localhost.
#
 TSAMySQLHost localhost
#
# (Only if TSADBModule is MySQL) Specifies the TCP port on which MySQL is
# listening. Not used if server is localhost. Default is 3306.
#
 TSAMySQLPort 3306
#
# (Only if TSADBModule is MySQL) Specifies the local UNIX domain socket
# on which MySQL is listening. Used only if MySQL is running on the localhost.
# Default is /tmp/mysql.sock.
 TSAMySQLUnixSocket /tmp/mysql.sock
#
# (Only if TSADBModule is MySQL) The MySQL user name to be used for connecting
# to the database. Default is tsa.
#
 TSAMySQLUser tsa_admin
#
# (Only if TSADBModule is MySQL) Name of the MySQL database for
# storing the tokens. Default is tsa.
#
 TSAMySQLDatabase TSA_Project
#
# (Only if TSADBModule is MySQL) Specifies whether a password must be entered
# at startup for connecting to the MySQL database. Default is Off.
#
 TSAMySQLPassPhrase On
```

ภาคผนวก ข

คู่มือการใช้งานโปรแกรม

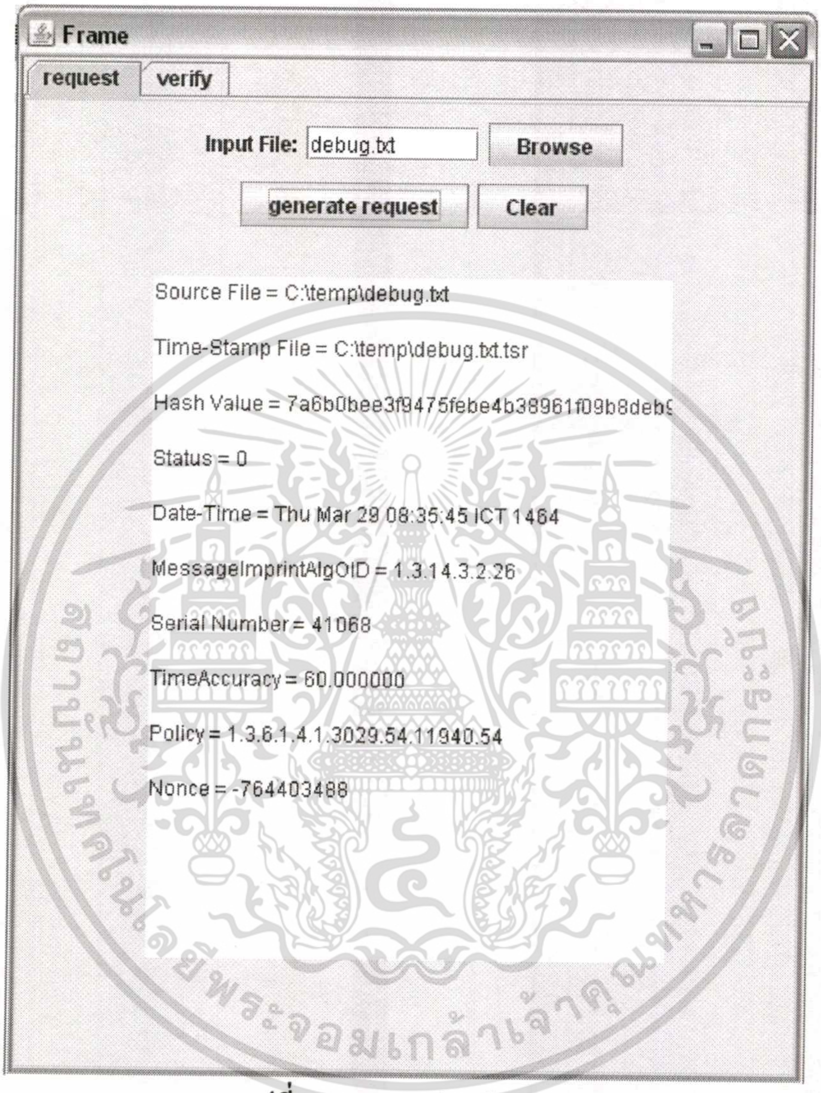
1. โปรแกรมประทับเวลา แบ่งการทำงานออกเป็น 2 ส่วน คือ
    1. ส่วนทำการสร้าง Time-Stamp Request
    2. ส่วนทำการตรวจสอบ Time-Stamp Token

การขอประทับเวลาเป็นหน้าจอสำหรับให้ผู้ใช้ทำขอประทับเวลาโดยผู้ใช้จะต้องทำการสร้างไฟล์ (Time-Stamp Request) และทำการส่งเข้าไปยังระบบ



รูปที่ ข.1 หน้าสำหรับผู้ใช้ขอประทับเวลา

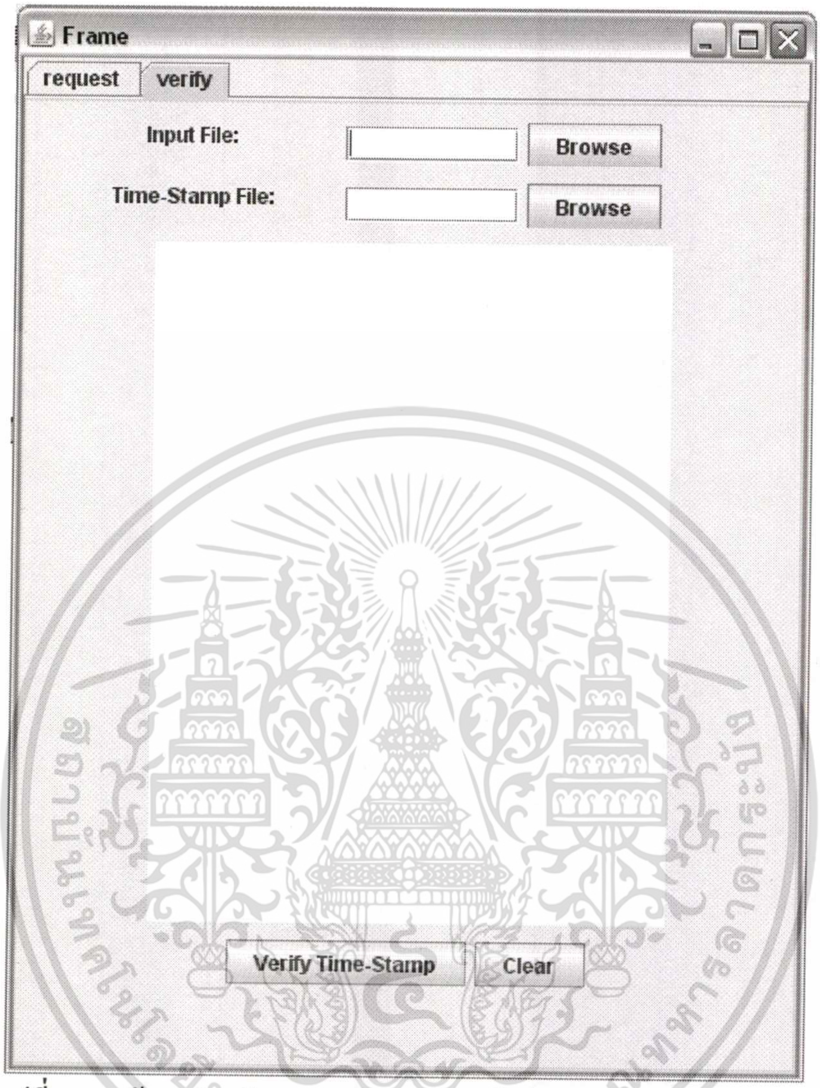
ผู้ร้องขอประทับเวลาทำการเลือกไฟล์ที่ต้องการประทับเวลาโดยกดที่ปุ่ม Browse หลังจากนั้นทำการกดปุ่ม Request Time-Stamp โปรแกรมจะทำการสร้าง Time-Stamp Request และส่งไปยังเซิร์ฟเวอร์ของผู้ให้บริการ



รูปที่ ข.2 ผลการประทับเวลา

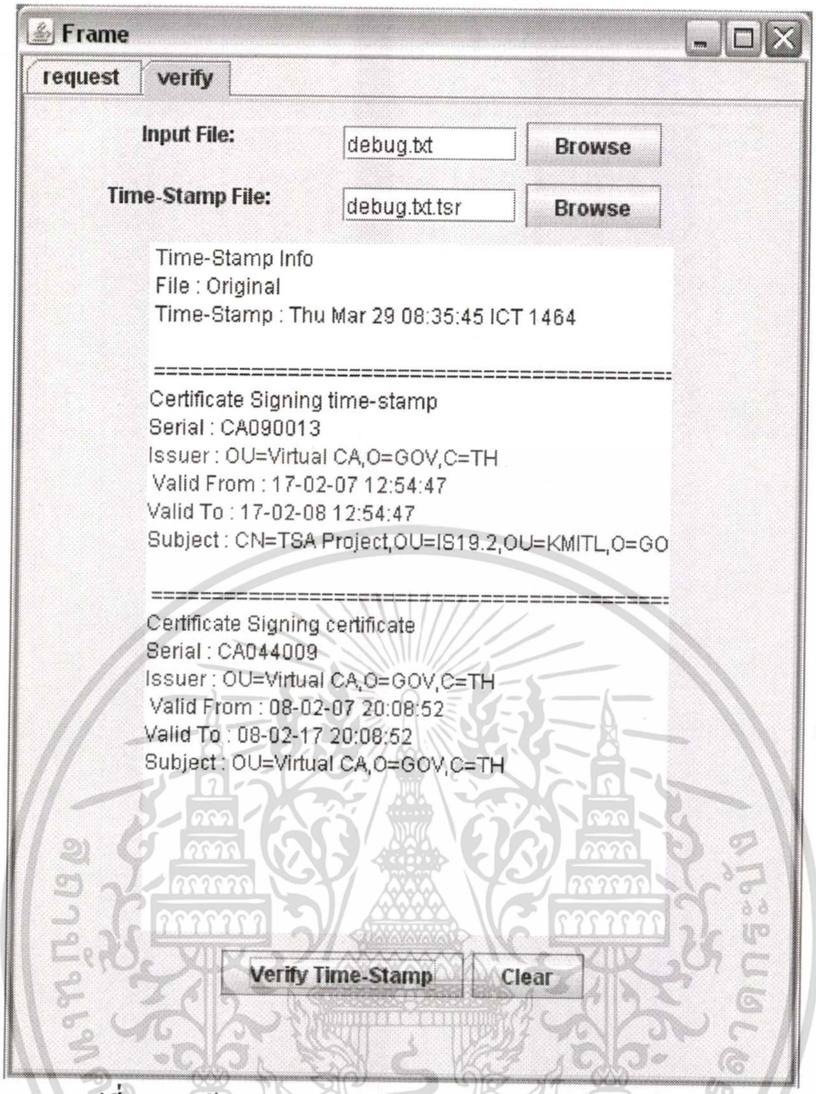
เมื่อเซิร์ฟเวอร์ทำการตรวจสอบความถูกต้องของ Time-Stamp Request แล้วจะทำการสร้าง Time-Stamp Token และทำการส่งกลับมายังผู้ร้องขอ ซึ่งไฟล์ Time-Stamp Token จะถูกบันทึกลงตามรายละเอียด Time-Stamp File ดังรูปที่ ข.2

การตรวจสอบข้อมูลประทับเวลาสำหรับให้ผู้ใช้ทำตรวจสอบข้อมูลประทับเวลา



รูปที่ ข.3 หน้าจอสำหรับตรวจสอบความถูกต้องของ Time-Stamp Token

เมื่อผู้ใช้ทำการตรวจสอบความถูกต้องของข้อมูลต้นฉบับ ต้องนำข้อมูลต้นฉบับและ Time-Stamp Token ที่คู่กันมาใส่โดยกดปุ่ม Browse เพื่อเลือกข้อมูลต้นฉบับและ Time-Stamp Token ตามลำดับ เมื่อได้ทำการเลือกเรียบร้อยแล้ว จึงทำการกดปุ่ม Verify Time-Stamp โปรแกรมจะทำการตรวจสอบความถูกต้องของข้อมูลต้นฉบับเปรียบเทียบกับ Time-Stamp Token



รูปที่ ข.4 หน้าจอแสดงรายละเอียดของ Time-Stamp Token

เมื่อทำการกดปุ่ม Verify Time-Stamp แล้วโปรแกรมจะทำการตรวจสอบและแสดงผลออกมา ซึ่งผลที่แสดงจะมีอยู่ 3 ส่วน คือ

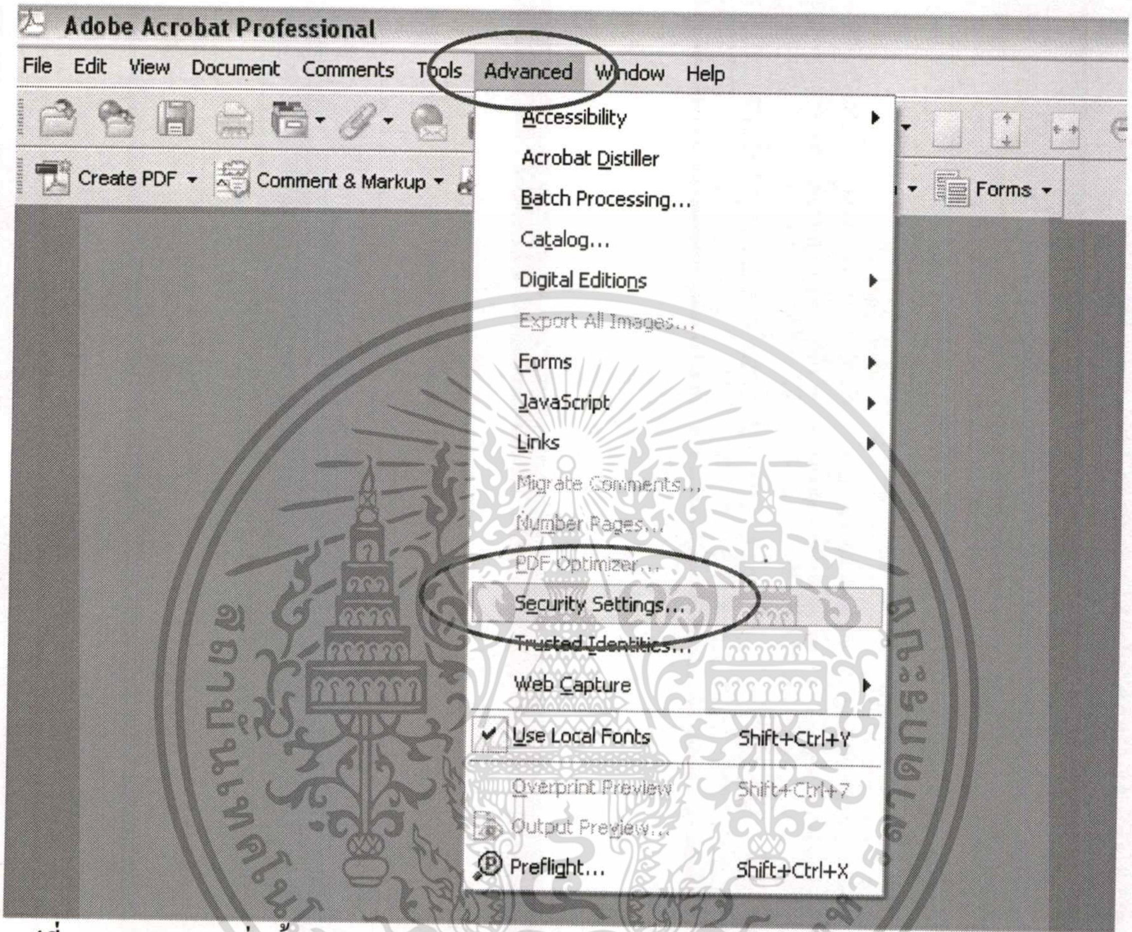
1. ส่วนแสดงความถูกต้องของข้อมูลว่ามีการเปลี่ยนแปลงหรือไม่ ถ้าไม่มีการเปลี่ยนแปลงการแสดงผลจะขึ้นว่า File : Original และแสดงเวลาที่มีการประทับเวลาไว้ แต่ถ้ามีการเปลี่ยนแปลงการแสดงผลจะขึ้นว่า File : Edited

2. ส่วนแสดงการตรวจสอบใบรับรองที่ใช้ในการประทับเวลาว่าอยู่ในสถานที่พร้อมใช้งานหรือไม่ ซึ่งจะขึ้นก็ต่อเมื่อ การตรวจสอบว่าระหว่างข้อมูลต้นฉบับกับ Time-Stamp Token นั้นแสดงผลว่าไม่มีการเปลี่ยนแปลง

3. ส่วนแสดงถึงใบรับรองที่เป็น Root ของใบรับรองที่ใช้ในการประทับเวลาว่าเป็นใบเดียวกันกับที่เราให้ความไว้วางใจหรือไม่ ซึ่งจะขึ้นก็ต่อเมื่อ การตรวจสอบว่าระหว่างข้อมูลต้นฉบับกับ Time-Stamp Token นั้นแสดงผลว่าไม่มีการเปลี่ยนแปลง

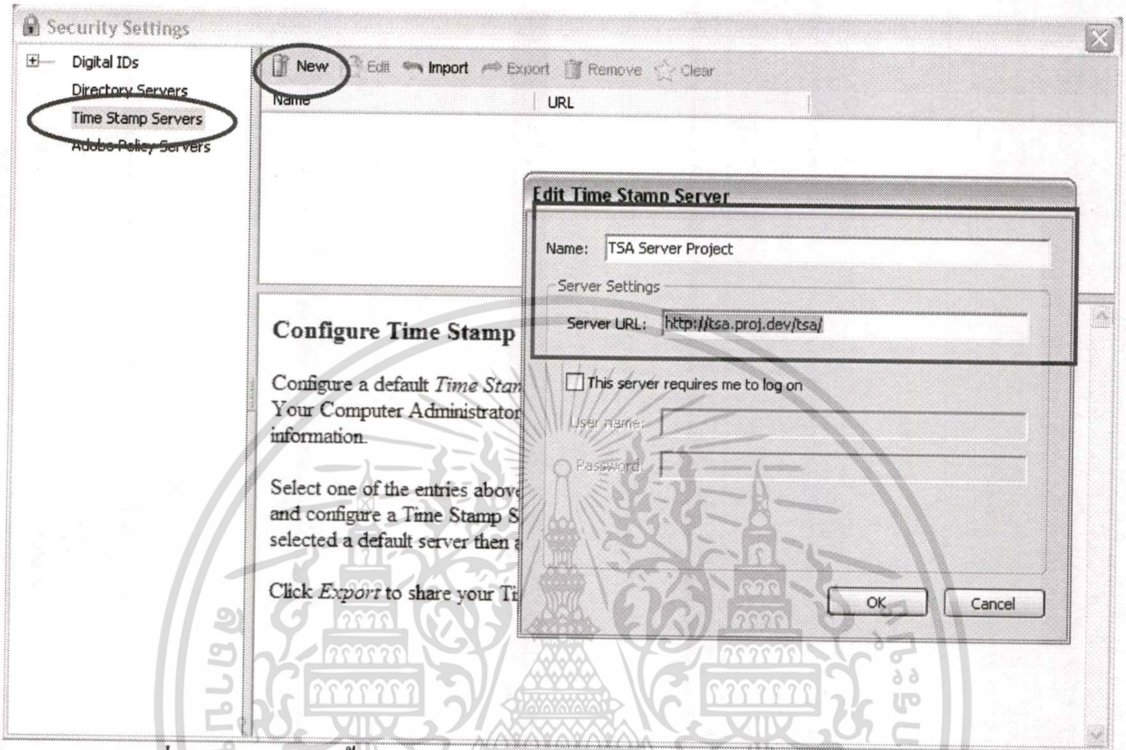
2. การตั้งค่าเพื่อให้โปรแกรม Adobe Acrobat เพื่อให้สามารถใช้ระบบผู้ให้บริการประทับเวลาได้

1. ทำการตั้งค่า Time-Stamp Server ให้กับโปรแกรม Adobe Acrobat โดยไปที่แทป Advance เลือกฟังก์ชัน Security Settings ดังรูปที่ ข.5



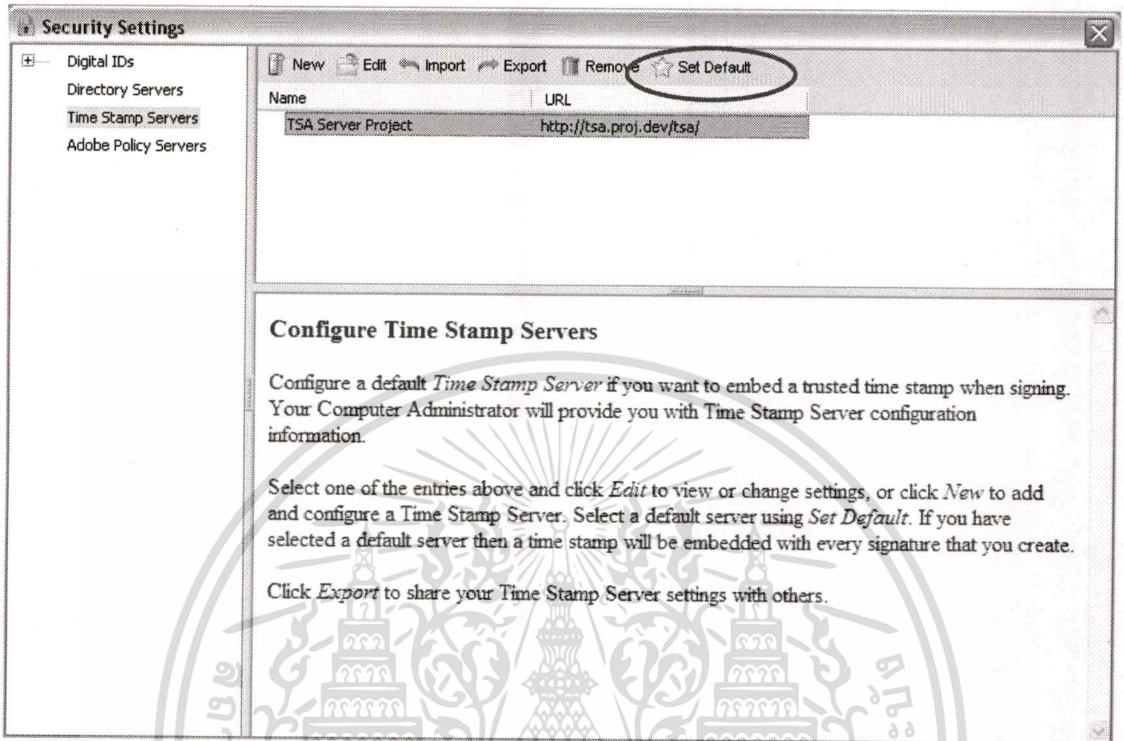
รูปที่ ข.5 แสดงเมนูเพื่อตั้งค่า Time-Stamp Servers ของโปรแกรม Adobe Acrobat Professional 7.0

2. ทำการเลือกที่เมนู Time Stamp Servers และกดปุ่ม New และป้อนค่าของ Time-Stamp Server เมื่อป้อนค่า Time-Stamp Server ถ้ามีข้อมูล username และ password เพื่อเข้าใช้ Time-Stamp Server ให้ทำการใส่ข้อมูลลงไปด้วย หลังจากนั้นกดปุ่ม OK ดังรูปที่ ข.6



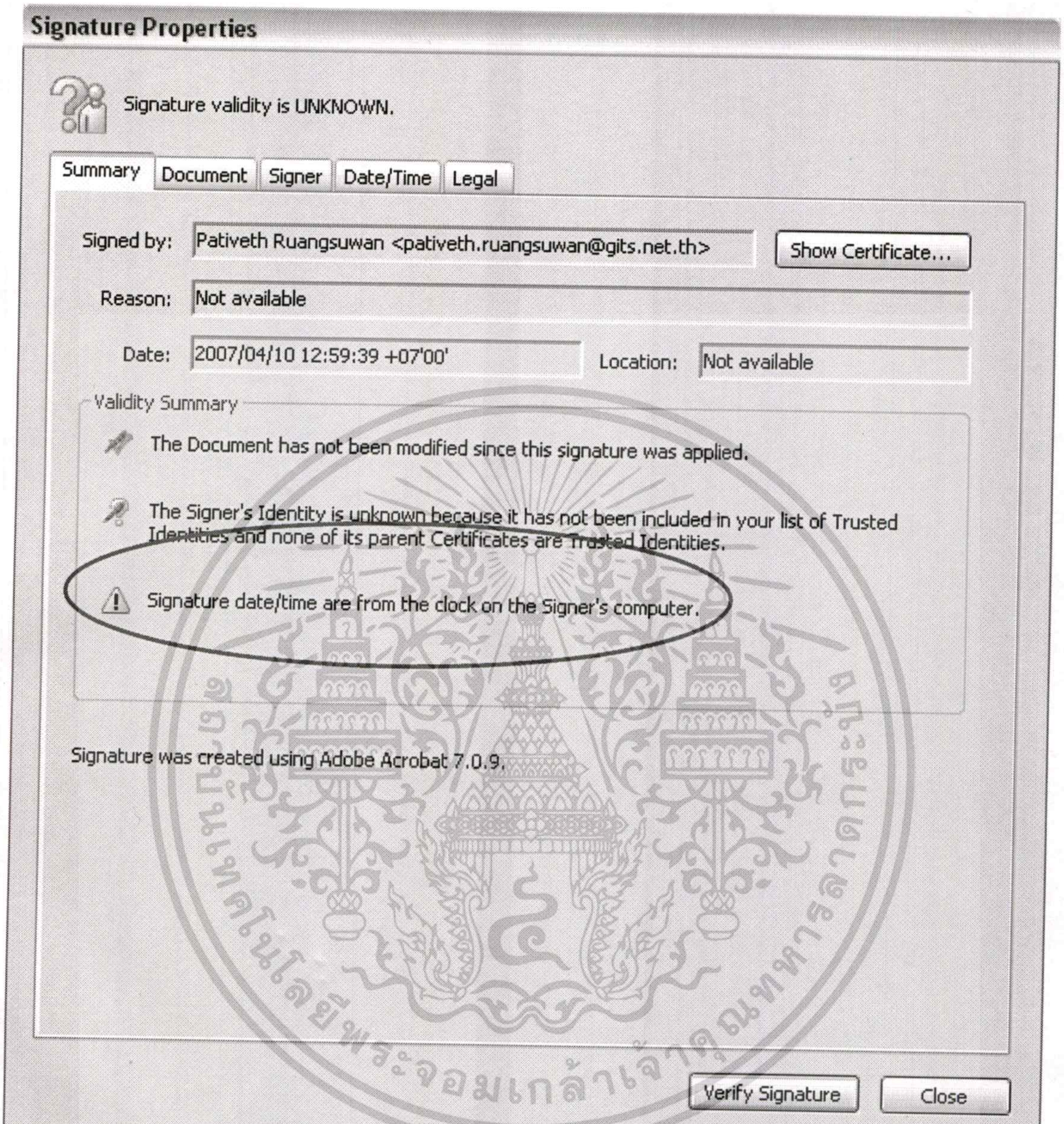
รูปที่ ข.6 แสดงการตั้งค่า Time-Stamp Server ให้กับโปรแกรม Adobe Acrobat

3. เมื่อทำการป้อนข้อมูล Time-Stamp Server เสร็จให้ทำการกดปุ่ม Set Default ดังรูป ข.7 เพื่อให้โปรแกรมทำการร้องขอประทับเวลาทุกครั้งที่มีการลงลายมือชื่อดิจิทัล

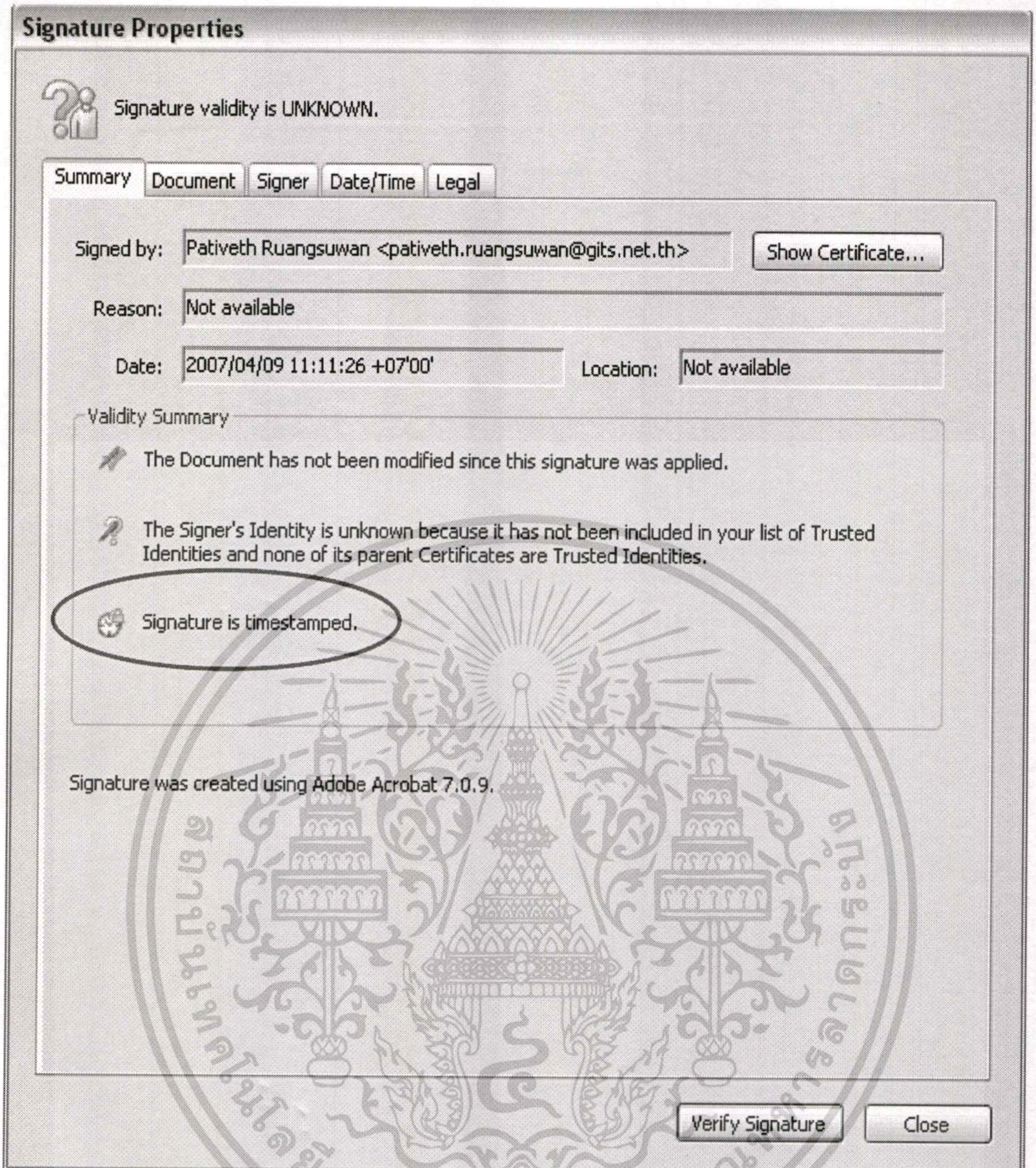


รูปที่ ข.7 แสดงการตั้งค่า Time-Stamp Server Default ให้กับโปรแกรม Adobe Acrobat

4. ตัวอย่างข้อมูลที่ไม่ได้ประทับเวลา ดังรูป ข.8 และข้อมูลที่มีการประทับเวลาดังรูป ข.9



รูปที่ ข.8 ข้อมูลที่ไม่ได้ประทับเวลาจากผู้ให้บริการประทับเวลาซึ่งจะใช้เวลาเฉพาะที่แทน



รูปที่ ข.9 ข้อมูลที่มีประทับเวลาจากผู้ให้บริการประทับเวลา

ประวัติผู้เขียน

ชื่อผู้เขียน	นายปฏิเวธ เรืองสุวรรณ
สถานที่เกิด	จังหวัดจันทบุรี
การศึกษา	ระดับปริญญาตรี
	วท.บ. (วิทยาศาสตร์บัณฑิต)
	สาขาคณิตศาสตร์
	มหาวิทยาลัยมหิดล
ประสบการณ์การทำงาน	สำนักบริการเทคโนโลยีสารสนเทศภาครัฐ